WALKOWIAK

AVENTURAS

Y COMO SE PROGRAMAN EN EL

ATARI®

600 XL / 800 XL / 130 XE

UN LIBRO EDITADO POR DATA BECKER S.A

WALKOWIAK

AVENTURAS

Y COMO SE PROGRAMAN EN EL

ATARI®

600 XL / 800 XL / 130 XE

UN LIBRO EDITADO POR DATA BECKER S.A

ISBN 84-86437-11-3

Déposito Legal B–26.877-85

Hecho el depósito que marca la ley.
I.S.B.N. 950-99088-3-5

Impreso en Argentina Printed in Argentina

Este libro ha sido traducido por Dña. Petra Scheffler arquitecta técnica y entusiasta del ATARI.

Distribuidor exclusivo:

UNIMPORT IBERICA, S.A.
Tels. (91) 247 31 26
247 31 21
Dos Amigos, 3
28015 MADRID

Advertencia Importante

Los circuitos, procedimientos y programas reproducidos en este libro, son divulgados sin tener en cuenta el estado de las patentes. Están destinados exclusivamente al uso amateur o docente, y no pueden ser utilizados para fines comerciales.

Todos los circuitos, datos técnicos y programas de este libro, han sido elaborados o recopilados con el mayor cuidado por el autor y reproducidos utilizando medidas de control eficaces. No obstante, es posible que exista algún error. FERRE MORET,S.A. se ve por tanto obligada a advertirles, que no puede asumir ninguna garantía, ni responsabilidad jurídica, ni cualquier otra responsabilidad sobre las consecuencias atribuibles a datos erróneos. El autor les agradecerá en todo momento la comunicación de posibles fallos.

INDICE

P R O L O G O

Este libro va dirigido a todos aquellos propietarios de un ATARI 400 XL / 800 XL / 130 XE, que se han cansado de los juegos de tiros y acción, y que ahora desean ocuparse de juegos más inteligentes.

Con este libro usted dispone de la llave para acceder al mundo de los juegos de aventuras, la llave de un mundo que hasta ahora sólo había rozado. Después de ojear las mas diversas revistas especializadas, habrá llegado varias veces a la conclusión, de que los programas allí publicados sólo convierten su ordenador en una cónsola de juegos.

Al estudiar el 'Aventuras - y como se programan con el ATARI', descubrirá en seguida que además de su aplicación para representaciones gráficas, el BASIC del ATARI también permite el tratamiento de textos, y aprenderá a usarlo.

Paso a paso vamos a desarrollar juntos una técnica en la programación de juegos de aventuras, cuya forma de presentación, tanto óptica como técnica, puede tranquilamente equipararse con las aventuras profesionales.

Aunque Vd. no sea un programador profesional, nosotros le daremos la base necesaria para llegar a entender los programas a través de unos textos , que bajo el nombre de Incisos, encontrará a lo largo del libro y con los que incluso los principiantes podrán seguirlo sin mayor dificultad.

A ello también contribuye la estructura propia de los programas, ya que despues de cada paso vera de forma práctica la exactitud de las correspondientes explicaciones.

Así pués, vamos a programar al mundo donde transcurre la aventura. A continuación buscaremos la forma de facilitar los movimientos al jugador o y también como proporcionarle los objetos que deberá utilizar.

Otro capítulo le iniciará en los secretos de la programación de aventuras. Encontrará también proposiciones adicionales sobre la manera de hacer la vida imposible al jugador y además,en este capítulo hallará instrucciones exactas con las que cualquier aventura confeccionada en texto puede ampliarse como aventura gráfica.

Y para que la configuración de gráficos no sea muy costosa, también dispondrá del listado de un programador gráfico que creará los subprogramas necesarios para usted.

Sin embargo, aparte de este programa auxiliar, con el listado del 'venturefix' también ponemos a su disposición un generador de aventuras, que le permitirá crear programas, aunque haya adquirido hoy mismo su ATARI y no tenga conocimientos de su programación.

Y en el caso de que usted prefiera antes que nada, jugar unas cuantas aventuras, en el último capítulo encontrará las aventuras en texto 'Fiebre del oro' y 'El castillo encantado', así como un listado de la aventura gráfica 'misión espacial'.

Supongo que preferirá ocuparse del verdadero tema de esta obra antes que leer un prólogo extenso. Así que, de momento,

sólo queda agradecerle la compra de este libro.

Y para finlizar, mi especial agradecimiento va dirigido al Sr. Dr. Achim Becker, que hizo posible la edición de este libro, así como a la Sra. Alicia Clees y el Sr. Claus Wagner, que me apoyaron infatigablemente en la redacción del mismo.

Recklinghausen,
Octubre 1984

Jorg Walkowiak

CAPITULO 1
- INTRODUCCION -

AVENTURAS, son unas experiencias extraordinarias de una persona, caracterizadas por una serie de situaciones extremas y peligrosas, que se graban para siempre en la memoria del aventurero.

Ejercen una gran influencia sobre el ser que está implicado en ellas y fascinan desde tiempos remotos incluso a los que no las han vivido directamente. La poesía se aprovechó de ellas y así se desarrollaron diversos géneros literarios del juego.

Siendo en un principio los trovadores los que relataron las hazañas de héroes lejanos, estas narraciones juglarescas fueron convirtiendose en poemas épicos, tales como 'Lanzelot' o el 'Parzifal' de Wolfram von Eschenbach.

Asimismo 'Don Quijote' y 'Munchhausen' entusiasmaron también a miles de personas.

Después de la época de la novela picaresca, el tema de la aventura se utilizó en narraciones de viajes (Defoe, Cooper, Karl May), hasta que su desarrollo llegó con las novelas negras y de ciencia ficción.

¿ Quién no conoce las novelas de aventuras mencionadas en la página anterior? quien no se ha entusiasmado con las aventuras de Old Shatterhand o Robinson Crusoe? -¿ Quién no se ha identificado con los protagonistas y se ha imaginado, su propia reacción en situaciones idénticas ?

Puede que usted sea una de las pocas personas que por razones de comodidad, nunca han leido un libro de este género. ¿Qué necesidad había de hacerlo?, se preguntara, ¡puedo conseguirlo de forma más cómoda!.

He visto en la televisión, Buffalo Bill y Tom Sawyer varias veces gracias a los directores de programación de nuestras emisoras. Entonces, ¿ por qué desaprovechar así mis horas de ocio, si también pueden hacerse otras cosas? Además, leer ya ha pasado de moda.

Cierto, probablemente tenga razón con el último argumento. Si antes las habitaciones de los niños estaban llenas de muñecas, cajas de construcciones, trenes eléctricos o libros infantiles, hoy estan llenas de modelos teledirigidos, cajas experimentales, videos, telejuegos y ordenadores personales. También aquí la técnica ha avanzado irresitiblemente, y es de agradecer si se utiliza con corrección.

Seguramente habrá usuarios de ordenadores que solo conocen batallas espaciales realistas y otros juegos de tiros y acción como única aplicación. Pero también estos usuarios algun día se darán cuenta, de que el proceso del juego es siempre el mismo. No se necesita pensar - tan sólo el ritmo del movimiento de las teclas de control varía un poco de un juego a otro.

Si se examina nuevamente el mercado del software, se llega a la conclusión de que los propietarios angloamericanos de ordenadores están mejor servidos; pues, además de

los conocidos Arcade-Games, los juegos de aventura también se comercializan en Inglaterra y America.

Las Aventuras, en inglés adventures, suponen seguramente la aplicación más inteligente del ordenador para fines lúdicos, aparte de los juegos estratégicos como ajedrez o Othello.

Estos programas permiten a propietario vivir sin grandes riesgos ni gastos, las más interesantes aventuras, en las que incluso lo imposible puede ser lo normal.

Imaginese un androide al que puede dirigir con instrucciones escuetas cualquier tipo de peligro, y al que puede dar instrucciones que ejecutará en su lugar y que decidiendo sobre la victoria o la derrota.

Ordene que vaya hacia una dirección determinada, y seguidamente le informará sobre lo que ve en su nuevo paradero, o le comunicará que desgraciadamente no existe ningún camino en la dirección deseada.

Hágale inspeccionar detenidamente las cosas que encuentre, ordene si le parece oportuno, que recoja estos objetos.

Finalmente, en vistas de conseguir el objetivo del juego, usted creerá haber encontrado una aplicación razonable para cada uno de estos objetos, haciendo reaccionar debidamente al androide. Disfrute su triunfo y, de que manera este juego le ha cautivado y alejado del mundo real.

Si ha cargado un juego de aventuras en la memoria de su ordenador, no sólo dispone de unas cuantas pantallas coloreadas para su juego, sino que se mueve en un mundo completamente equipado, un mundo que sólo esta esperando, que usted descubra y resuelva sus secretos y enigmas.

Un buen juego de aventuras le permitirá actuar como en la vida real, esto quiere decir, que acciones y reacciones se relacionan de forma habitual, y que experiencias de la vida también resultan útiles en el juego por ordenador.

Descubrirá que incluso una acción que se realiza diariamente, como el abrir una puerta, puede presentarle dificultades que se anuncian con la emisión de un mensaje tal como '¡No tengo la llave correspondiente!'.

Incluso puede ser necesario recurrir a la magia para obtener la respuesta 'O.K. - la puerta está abierta.', puesto que sólo la fantasía del programador establece los límites entre lo posible y lo imposible.

Por otra parte, el juego de aventuras también puede utilizarse como medio de enseñanza. Así se puede confeccionar por ejemplo, una aventura equipada con los datos de nuestro sistema solar, que nos permita explorar el espacio cómodamente sentados en nuestro sillín favorito.

¿ Qué método podría ser el más adecuado para enseñar tanto a niños como a adultos deseosos de adquirir conocimientos generales o específicos?

Sea como sea, tanto si un juego de aventuras ha sido escrito para servir como programa de juego, como si sólo se ha pretendido utilizarlo para el entretenimiento, ya no es suficiente tener una buena capacidad de reacción para poder superarlo.

Necesitará un entendimiento agudo y una capacidad de razonamiento lógico para poder superar los numerosos peligros que se presentan en la vida de un aventurero:

¿ Cómo quiere pescar un pez para que no tenga que morir de hambre, si sólo dispone de sus manos para hacerlo?
¿ Cómo puede evitar que el oso hambriento le considere una suculenta merienda?
¿ Cómo puede cruzar un rio tormentoso què además está poblado de prehistóricas fieras salvajes?

Todos estos problemas pueden solucionarse razonando un poco, sólo necesita disponer de las ideas adecuadas y tener mucha fantasía:

Su misión consiste en conseguir introducirse en una determinada casa, cuya entrada está asegurada por un portal enrejado. Fijandose bien usted divisa una llave a unos cuantos centímetros detras de la verja, pero sus brazos son demasiado cortos para poder coger la llave a través del portal.

¿Cómo puede entrar en la casa?

Saltar la verja no es posible, pero a unos pocos pasos de la casa, usted vío unos árboles. En algún momento encontró además un chicle.

¿ Ya se ha dado cuenta?

Nosotros también hemos tardado un poco en encontrar la solución a este problema, naturalmente nadie nos ha indicado de forma tan clara los objetos que pueden ser útiles:

Sólo era necesario dar una vuelta alrededor del árbol y arrancarle una rama. Con ella se había de retroceder

hacia la entrada de la casa, mascar bien el chicle, pegarlo a un extremo de la rama, y hacer pasar a esta a través de las rejas del portal y se tocando la llave, con lo que ésta se adheriría al chicle. Retirando el palo con cuidado, se podía conseguir finalmente la llave.
¡Qué suerte el que esta llave abriera el portal después!

Esta ha sido una solución elegante, pero no demasiado fácil, aunque el programador ya la había planeado de esta manera en los trabajos de confección del juego. Sin embargo, yo debería añadir, que en las instrucciones que acompañan a la aventura se prevé un mínimo de experiencia de juego.

Quizás el ejemplo expuesto anteriormente ya le haya afectado el ánimo de aventurero. Pero si el poder de la palabra escrita aun no ha conseguido fascinarle, lea detenidamente la "Historia interminable".

¿Qué posibilidades de superación resultan en un sistema de ordenador de la implementación de buenas novelas?

Quizás algún día resulte necesario ampliar la definición del término "aventura" en nuestro diccionario:

> En el siglo 20, la técnica moderna de la informática proporcionó la novela de aventuras para participar en ellas a través del juego. Como punto de partida sirvió "Adventureland" de Scott Adams. Le seguieron otras muchas aventuras que evolucionaron hasta una perfecta simulación del mundo. Numerosos autores de novelas de prestigioso nombre (por ejemplo Michael Crighton) se dieron cuenta a tiempo de las nuevas posibilidades, y con sus obras consiguieron para mucha gente el equilibrio necesario en la vida cotidiana.

HISTORIA Y DESARROLLO DE LAS AVENTURAS

Después de que la primera máquina calculadora se convirtiera en un ordenador, en el mundo de la programación se realizaron los más variados esfuerzos para enseñar a los ordenadores las formas humanas de pensar y de actuar.Finalmente en el año 1966, finalmente, Joseph Weizenbaum del Massachusetts Institute of Technology alcanzó con su programa ELIZA, no sólo los expertos, sino también, público en general.

ELIZA, que actualmente también se puede adquirir en versiones reducidas para cualquier tipo de microordenador, simula a un psiquiatra e imita su técnica conversacional típica. En ensayos realizados con personas a modo de prueba, éstas se mostraron muy sorprendidas, por haber confiado sus problemas personales a una máquina.

El desarrollo ulterior de la inteligencia artificial ofreció a los científicos técnicos un programa implementado en un PDP-10: ADVENTURE - Colossal Cave. Para variar, estos sabios podían investigar en otro campo, que era el de los tesoros de un mundo oscuro dominado por la magia. Colossal Cave disfrutó de una gran popularidad en medios técnicos y, en verano de 1979, se presentó finalmente al público. La empresa MICROSOFT, conocida actualmente en todo el mundo, publicó Microsoft's Adventure, una versión en diskette de estas aventuras primitivas destinadas al TRS-80, el más popular microordenador en América en ese momento.

Sin embargo, las aventuras se conocieron ya en el año 1978 por un joven americano llamado Scott Adams, que con su ADVENTURELAND crea la base para el nuevo ADVENTURE INTERNATIONAL.

También en este juego se basa en encontrar tesoros dentro de un mundo irreal, aunque naturalmente dificulta

con la presencia de dragones, laberintos, rios de lava y demás singularidades, habiendo empezado inocentemente en un bosque. Pero una vez se ha orientado y sabe hacia qué dirección debe dirigirse, y si además se encuentra la entrada que conduce hacia el imperio subterráneo, entonces...

La demanda por ADVENTURELAND era inmensa, seguramente debido a que todavía no existían juegos con la calidad propia de los juegos de salones recreativos. Debido a este gran éxito, el ADVENTURELAND se convirtió rápidamente en una serie de 12 aventuras, procurando siempre mantener despierto el interés del cliente con la elección cuidada de los temas. ¿Ha soñado alguna vez con enfrentarse al conde Drácula? Pues puede hacerlo en "The Count". ¿o prefiere tropezar con un saboteador en una central de energía atómica? No hay ningún problema, ''Mission Impossible''se lo permite.

Durante este tiempo, en el mercado americano de software apareció en el mismo nivel un tipo de programa que se llamó ''INTERACTIV FICTION''. Estos programas debían representar un nuevo género literario que, tal como indica su nombre, serán libros para colaborar de forma activa. En principio, el jugador sólo es un simple lector, que ha sido introducido con precisión en la acción. Numerosas páginas se muestran en la pantalla, informado al lector sobre los antecedentes, situaciones actuales, causas, etc., y que pueden ser leidas igual que un libro.

Así, "Local Call For Death" comienza como una historia criminal: tres hombres se encuentran en su club en el transcurso de la noche, uno de ellos es llamado al télefono. Es el sobrino visiblemente irritado, que pide dinero al tío. Sin embargo, el tío se niega, sabiendo por experiencia, que este dinero servirá para efectuar apuestas en las carreras. A la mañana siguiente, se

se entera del suicidio de su sobrino.

A partir de este momento, el jugador entra en acción representando el papel principal, intenta aclarar el presunto suicidio conversando directamente con los protagonistas.

La característica más destacada de estos programas es la existencia de un claro y único objetivo, como en este caso lo es el esclarecimiento de un asesinato.

Para conseguirlo, los jugadores no deben servirse de objetos de cualquier clase sino de preguntas astutas. ¿Qué será más efectivo?: ¿leer un diálogo en un libro, seguir un interrogatorio en una película, o disfrutar de la satisfacción de haber dado con el criminal ayudado por la propia capacidad de combinación?

Precisamente esta facultad de diálogo es el principio que los 'Interactiv Fiction' usan para estimular el juego. Una muestra ejemplar de ello es "Encounter in the Park", aparecido también en Adventure International.

En este juego uno encuentra a la mujer de sus sueños durante el paseo matutino; nuestro objetivo consistirá en seducirla hasta conseguir que nos de su SI definitivo. El programador pone a disposición del jugador un amplio surtido de técnicas persuasivas que le permitirán alcanzar sus deseos; es realmente divertido ver como reacciona la señorita ante determinadas proposiciones.

Lo peor que puede ocurrirnos si estas proposiciones son poco honestas es, que ella se aparte de nosotros indignada y que se nos informe de que, debido a la decepción ocasionada, pasaremos el resto de nuestra vida encerrados en un convento.

Como ya se ha mencionado, el interés por todos estos tipos de aventuras en texto fue creciendo progresivamente. Por ello, cada vez más empresas de software editaron copias de estos juegos con distinta calidad

Estimulados por la creciente presión que ejercía la competencia, posteriores fabicantes de software crearon nuevos juegos de aventuras.

Aparecieron los primeros "LABYRINTH - ADVENTURES", donde el jugador tenía la misión de salir con vida de un edificio. El juego solía terminar casi siempre con el jugador perdido sin remedio en el laberinto representado en perspectiva en la pantalla.

La próxima etapa de este desarrollo fueron los "QUEST - ADVENTURES", que casi siempre fueron implementados como aventuras de tiempo real. Además, este miembro de la gran familia de las aventuras produce al jugador a un estado de stress, pues al aparecer algún ladrón u otros malhechores, el jugador no pulsaba inmediatamente la F para abrir fuego y seguia pulsando continuamente unas teclas cualquiera o incluso combinaciones de teclas, que provocan que su espada apunte hacia una determinada dirección, el juego termina relativamente pronto por el excesivo deterioro del personaje principal.

Desgraciadamente, la forma en que estos juegos están concebidos limita considerablemente el campo de acción del jugador, ya que sólo dispone de una restringida capacidad de actuación con acciones de lucha,negocio,compra o huida, que le son ofrecidas utilizando el menú, condicionando el apelativo QUESTion (pregunta).

La siguiente etapa en el desarrollo de los juegos de aventuras fue posible gracias a la baja sufrida por los precios del los procesadores gráficos y de los elemontos RAM. De esta manera, los fabricantes de ordenadores ya no tuvieron dificultades en proporcionar al usuario una económica capacidad gráfica para ordenadores, capacidad que hasta el momemto sólo era posible con equipos mucho más grandes y, en consecuencia, más caros.

Así comenzó en el año 1982 (para los usuarios de Apple un

poco antes) el nuevo apogeo de estos juegos con la venta del primer "GRAFIKADVENTURES" o aventuras gráficas. Si varias propietarios de ordenadores habían rechazado hasta este momento los juegos en texto, ahora empezaban a mostrar interés por su gran variedad de imágenes en color. Incluso eran útiles para conseguir que la familia conocedora de informática exclamara un sorprendido ''¡Oh, ue bonito!'' ante la pregunta ''¿Qué puede hacer tu ordenador?''

Naturalmente el valor propio del juego y su objetivo no habían cambiado en nada.

Sin embargo, de repente se podía acceder a otra clase de compradores, lo cual ya era una buena razón para lanzar al mercado todas las aventuras conocidas durante años pero disfrazadas de otro modo: habían desaparecido las explicaciones exhaustivas y, en su lugar, se habían visualizado en la pantalla las situaciones momentáneas .

De todas formas no esta claro si los programas resultan más fáciles de manejar. Yo personalmente prefiero que un mensaje como

Estoy en el bosque. A mi alrededor hay muchos árboles. Entre dos robles veo un agujero.

me indique informaciones importantes, a ver en su lugar, sólo unos cuantos árboles. Seguramente, en el primer momento los considerará como una parte más del bosque, por lo que seguramente seguirá adelante sin detenerme a examinarlos más de cerca, y sin distinguir por supuesto, el hoyo en forma de pequeña mancha negra que hay entre dos de ellos. Como es lógico, precisamente en este hoyo es donde se encuentra una parte del tesoro buscado, o un objeto cuya posesión se hace imprescindible para poder llegar con éxito al final de la aventura.

CAPITULO 2
- EL CONCEPTO -

En los Próximos capítulos de este libro crearemos un juego de aventuras conjuntamente. Por supuesto, exigiremos que nuestra obra tenga una presentación lo más profesional posible, hecho que impedirá prescindir del análisis de determinadas características y rasgos de algunos de los programas que se encuentran en el mercado.

Acto seguido, nos preocuparemos de cómo debemos programar cada una de las funciones en BASIC.

Para ello intentaremos conseguir que las rutinas tengan una estructura tal, que no sólo permita utilizarlas en una aventura determinada, sino que también nos sirva en cualquier otro programa de juegos sin requerir grandes modificaciones.

LA PRESENTACIÓN

Ya hemos mencionado, que una aventura pretende situar al jugador en un mundo extraño, en un mundo donde se hace posible lo imposible. En este mundo es donde vivirá las aventuras; por consiguiente, deberá tener capacidad de movimiento y actuación, y deberá ser informado sobre los resultados de sus esfuerzos para que pueda proseguir con otras acciones.

Sin embargo, una actuación razonable y precisa sólo es posible si se hace uso de los sentidos. Esto significa, que se deberá proporcionar al jugador, de forma adecuada, todas aquellas informaciones que el aventurero percibe en la vida real a través de la vista, oidos y tacto.

Puesto que hasta hoy no se ha conseguido conectar el cerebro humano directamente al ordenador, las respuestas a las preguntas ''¿Dónde estoy? - ¿Qué veo? - ¿Hacia dónde puedo ir? - ¿Qué siento?'' deberán mostrarse en forma de frases cortas pero completas.

Sin embargo, la adaptación al ambiente que rodea al jugador pocas veces ocurre de manera consciente ya que el contenido de los distintos lugares son registrados en décimas de segundo sin haberlos pedido específicamente. Por ello no se le puede pedir al jugador de la aventura que se entretenga en extensas explicaciones sobre el lugar en que se encuentra.

Pocas veces ocurre de forma consciente ya que los contenidos de los lugares son registrados en décimas de segundo sin haberlo pedido específicamente. Es por ello que no se le puede pedir al jugador que se entretenga en extensas explicaciones sobre el lugar que acaba de pisar.

Incluso no siendo necesarios más de cinco o diez segundos en la lectura de algunas líneas de descripciones, la concentración requerida detendría la buena marcha del programa e impediría que el jugador se identificase con el personaje del juego entergandose a las ilusiones de un mundo imaginario.

Para evitar este problema, se limita la salida de texto a un mínimo aceptable de informaciones, que se presentan, si no se trata de aventuras gráficas, en dos zonas claras de la pantalla, cada una de las cuales tiene una determinada función.

De esta manera, la parte superior de la pantalla nos informa sobre el lugar en el que nos encontramos y sobre lo que divisamos, mientras que la parte inferior se destina a la comunicación con el personaje principal que actúa por nosotros. Es aquí donde indicaremos nuestras instrucciones y nos enteraremos de lo que ocurre a consecuencia de ellas.

Una típica imagen de pantalla podría ser:

```
Estoy en un bosque oscuro.
Veo muchos árboles viejos.

Puedo ir al Norte, Este, Oeste.
--------------------------------

¿ Qué debo hacer?
```


Si lo que pretendemos es avanzar en el juego lo más rápido posible, sería erroneo por nuestra parte continuar la marcha en cualquiera de las tres direcciones.

Lo más razonable que puede hacer un jugador de aventuras en situaciones que desconoce, es investigarlo todo a conciencia. Así, las posibles respuestas a la instrucción ''Investiga árbol'' podrían ser como: ''Se trata de un roble.'' o ''Aquí también se propaga la extinción de árboles.''

Estas respuestas no nos solucionan ningún problema, ya que sólo parecen demostrarnos, que no siempre es tan interesante el investigarlo todo. Pero, ¿cómo saberlo con antelación a las respuestas? También podría haberse dado una respuesta como ''En un tronco veo un orificio'' Quizás se trate de un viejo nido de un pájaro carpintero en el que se encuentra precisamente una parte del tesoro buscado. ''Investiga orificio'' podría descubrirlo.

En el caso de que no hubiésemos encontrado ningún punto de referencia, ni a la instrucción ''Investiga bosque'' - podría tratarse de un bosque mágico - deberíamos preguntarnos en que dirección queremos ir.

¿Cómo se comportaría usted en una situación parecida, sin mapa ni compás, desamparado y sólo, lejos de cualquier camino ?

Posiblemente subiría a un árbol para ver que hay en las inmediaciones.

Por consiguiente, debería entrar: ''Sube a un árbol'', y si no recibe un mensaje de la forma ''¡No soy lo bastante deportista para ello!'', se modificará la imagen de su pantalla:

```
Estoy en la cima de un viejo roble.
Veo montañas en el Norte,
un lago en el Este,
del Oeste sube humo.

Puedo ir hacia abajo.
--------------------------------
```

SUBE A UN ARBOL

¡O.K.!

¿Qué debo hacer?

Con esto la cosa mejora , ya que tras la entrada de "A" para indicar abajo, sólo nos queda solucionar el problema sobre la zona que visitaremos primero.

El mensaje ''¡O.K!'' debe informarnos de que la instrucción entrada ha sido entendida y ejecutada. Si no fuera así, habríamos recibido por ejemplo, ''¡No entiendo el verbo!'', con lo cual el programa nos induciría a cambiar la forma de la expresión. Lo mismo sirve, naturalamente, para el objeto de nuestra instrucción.

Sin embargo, también podemos suponer otras dos respuestas. Por un lado recibiremos a menudo, el mensaje ''I must be stupid, but... -¡No entiendo lo que quieres decir !'', con lo cual el programa nos comunica que encuentra nuestra entrada muy poco razonable, como lo sería sin duda en caso de entrar una instrucción como ''Mata árbol''. Esto significa que esta acción no esta prevista en el juego.

Algo parecido ocurre con la segunda respuesta estandar ''You can't do that ... yet. - Esto no es posible de momento.'', que nos indica que las palabras empleadas pertenecen al léxico programado, pero que la instrucción carece de sentido en este momento: el programa del juego sólo entiende "Coge llave", si aparece una llave en la aventura y sin embargo, éste mensaje pierde sentido mientras no se encuentre una llave en el mismo lugar donde éste situado el jugador.

A pesar de ello podemos tener éxito, puesto que si obtenemos esta respuesta vamos por buen camino; sólo falta cumplir algunas condiciones más, que se necesitarán para ejecutar la instrucción. Así podrá abrirse una puerta cerrada si conseguimos el instrumento apropiado para ello. Si por el contrario, la respuesta a "Abre puerta" hubiese sido "No entiendo lo que quieres decir", todo esfuerzo sería inútil.

PREMEDITACIONES

Ahora ya estamos familiarizados con la estructura general de los juegos de aventuras. Ya tenemos una idea de como debe presentarse nuestra aventura en la pantalla y qué informaciones deben prepararse. A continuación vamos ocuparnos más de cerca de cada uno de los elementos que componen nuestro sistema de aventura, y elaborar una concepción que se realizará en el capítulo siguiente.

Lugares de la aventura:

El mundo de la aventura se compone de los lugares concretos de cada juego, esto significa, que cada uno de los sitios accesibles por el jugador que atraviesa la aventura recibe el nombre de lugar. Su tamaño no tiene ninguna importancia. Puede ser, como en nuestros ejemplos anteriores, un bosque o la cima de un árbol, pero también puede tratarse del interior de un coche o de un armario.

Cada lugar se distingue para el jugador a través de sus descripciones y de su disposición geográfica. En cuanto a la técnica del programa, van a distinguirse por su número de lugar, como veremos más adelante.

Todos estos lugares son accesibles para el jugador, por lo

que deben estar comunicados de manera apropiada. En cada aventura se permite el movimiento hacia los cuatro puntos cardinales Norte, Sur, Oeste y Este; algunos incluso admiten el movimiento en vertical.

Naturalmente no debe ocurrir, que el jugador abandone el lugar 5 en dirección Oeste, entrando en el lugar 6 por el Este y, si desea volver a entrar en lugar 5, no encuentre ningún camino en dirección Este, o peor todavía, si con un movimiento hacia abajo vuelve a acceder al lugar 5.

Para prevenirse de la crítica de aventureros experimentados, también en este caso la excepción confirma la regla, pero ¿que sería de una aventura sin un laberinto mágico?

Objetos

Otros elementos típicos de cada lugar son los objetos que se encuentran en él. Debemos colocar todos los objetos correctamente en los lugares o, dicho más correctamente, a cada objeto le corresponde un lugar. En esta asignación evitamos automáticamente, que algún objeto exista doblemente.

Otro problema se nos presenta con la colocación de aquellos objetos que no pueden encontrarse en la aventura al comenzar el juego. ¿Dónde podemos guardarlos hasta que deban hacer su aparición?

Supongamos que nuestro héroe entra en un lugar y que, como objeto visible, sólo encuentra una vieja caja de madera cerrada.

Reaccionando a la instrucción del jugador "Abre la caja", todas las reglas de la lógica imponen, que la vieja caja cerrada desaparezca de la pantalla, y que en su lugar aparezca una vieja caja abierta posiblemente llena de monedas de plata.

Así pues, la solución a este problema no nos presentará demasiadas dificultades. Instalaremos en nuestra aventura un lugar adicional, el "almacen", donde guardaremos todos los objetos no utilizados de momento. Si no programamos ninguna comunicación hacia este lugar, él jugador tampoco podrá acceder nunca a él ni entrar en contacto con los objetos correspondientes de manera indeseada.

En el próximo capítulo veremos lo fácil que es programar el almacén y también el completo mundo de la aventura. Además veremos también en el transcurso del libro, la necesidad de disponer de otros lugares especiales.

Entradas

Tal como resulta del texto precedente, las entradas del jugador suelen estar compuestas de un verbo y un objeto. Esto significa que nuestro programa deberá comprobar primero, si una entrada esta admitida o no, para lo cual será necesario separar el verbo del objeto. Si el jugador se ha equivocado, o si el verbo utilizado no está previsto, deberá efectuarse un mensaje consecuente. Cuando nuestro juego de aventuras haya encontrado en su léxico el verbo entrado, deberá comprobar el objeto de la misma manera.

Si ambos términos estan definidos, el programa podrá reaccionar de la forma impuesta.

Por lo demás, el programador debe proceder con especial precaución en la confección de este léxico. Deberá preveer todas las posibles entradas que cualquier jugador puede efectuar a lo largo del juego.

Lo difícil que puede resultar precisamente esta empresa, me

lo demostró hace poco uno de mis amigos, que nunca antes había jugado una aventura. Después de estudiar la imagen titular junto con las instrucciones más esenciales, este, quería resolver todos los problemas instantaneamente con la instrucción "Encuentra tesoro", una entrada que yo, como programador, no había previsto por supuesto.

también se hará necesaria la introducción de sinónimos, ya que si el jugador se entretiene demasiado tiempo para exponer sus deseos , perderá rápidamente el interés en ello.

Sin embargo la elección de estos términos puede ser un problema para nosotros como españoles. Expresiones que en inglés no representan ningún problema ''Climb tree'', ''Shoot gun'', ''Drop box'', ''Look'', no quedan tan claras en castellano: ''Trepa árbol'', ''Dispara rifle'', ''Pon caja'', ''Mira''.

Mensajes

La primera impresión en pantalla, después de las descripciones del lugar, informa al jugador sobre los puntos cardinales en cuyas direcciones puede volver a abandonar el lugar de acción.

Por otra parte, también se debe aclarar al jugador la ejecución de cada acción, así como una posible reacción que pueda producirse en consecuencia. En el caso más simple, esto sucede a través de "O.K", pero en la mayoría de los casos se añade la emisión de mensajes adicionales. A veces, las entradas más diversas pueden requerir la misma repuesta. Precisamente por esta razón, estas respuestas deberían confeccionarse en una lista y ser estandarizadas. Así por ejemplo, pueden aparecer con cierta frecuencia respuestas como ''No tengo tanta fuerza.'', o ''¡Esto no tiene ningún sentido!'' y ''¡No entiendo lo que quieres decir!''.

FUNCIONES DE LAS AVENTURAS

En un juego de aventuras encontramos un promedio de 30 - 50 lugares diferentes, y en cada lugar hay uno o dos objetos. Estos objetos pueden contener por su parte, cualquier cosa, y como en una aventura no se sabe nunca exactamente lo que se necesitará un par de jugadas más adelante, muchos jugadores tienden a llevarse todo lo que les cae en las manos , que no sea demasiado pesado para llevarlo a cuestas.

Con todo ello resulta bastante difícil acordarse de cuanto llevamos, ya que nos hemos vuelto muy cómodos y naturalmente, no tomamos notas.

Los programadores se dieron cuenta muy pronto de este hecho, y así actualmente cualquier aventura muestra en pantalla todos los objetos acumulados, despues de entrar la instrucción 'INVENTARIO'.

Si se trata de un juego de aventuras simpático o escrito especialmente para principiantes, podemos pedir consejos en situaciones complicadas, ya que entonces está implementada la instrucción 'HELP' - 'Socorro'.

Al contrario de la entrada estandarizada compuesta por verbo y objeto, aquí se trata, al igual que en balance, de una instrucción de una palabra, que en este caso no provoca siquiera una reacción.

Pero esta instruccion puede tener algunas consecuencias:

''Yo lo examinaría todo'' nos indica nuestra negligencia; el mensaje ''Aparece una figura mágica y escribe la frase 'Abrete sésamo' en la arena'', en cambio, nos hará dar un salto de alegría en una situación complicada.

El asunto adquiere peor aspecto si hemos de pagar por la ayuda: 'Aparece un ogro y dice que perdemos 10 puntos por su ayuda. Quiere saber si necesitamos ayuda'.

Probablemente aparecerán nubes oscuras en el cielo, si en la entrada de socorro obtenemos la respuesta siguiente: ''¡Socorro no está previsto en esta aventura !'', o si sólo se repiten instrucciones generales de juego como respuesta.

Es naturalmente molesto, cuando no se puede avanzar en una situación a pesar de todos los esfuerzos realizados. Sin embargo, nadie debería considerar los dos últimos ejemplos como actos de malicia; si la función de socorro se admitiera continuamente, algunos jugadores muy inteligentes resolverían la aventura en menos de 30 minutos, pero sin llegar a conocer nunca la fascinación que estos programas comportan.

Por esta razón, algunos programas cuidan el uso de esta función , niegan finalmente cualquier ayuda o conducen al jugador a pistas falsas, si se pide socorro con demasiada frecuencia.

CAPITULO 3
- LA REALIZACION -

Antes de empezar con la programación, quisiera darle algunas referencias de tipo más bien técnico, que deben contribuir a que no se pierdan las ganas de trabajar con este libro por la aparición de algún error.

Seguramente, las próximas líneas de programa no serán las primeras que usted entre con esfuerzo a través del teclado, y habrá adquirido la experiencia de que siempre vuelven a aparecer errores al teclear.

Hay determinadas líneas DATA que parecen estar predestinadas a errores de esta clase, especialmente si no contienen más que cifras, como en este caso.

Por otra parte, es imprescindible que algunas líneas sean introducidas exactamente tal como están imprimidas aquí. Esto es válido, sobre todo, para el número de blanks (espacios) en los strings (variables de cadena).

Para proporcionarle una efectiva posibilidad de control, todos los listados de este libro se han imprimido de la manera que usted debería verlos en la pantalla después de la entrada de LIST.

Cuando tenga la impresión de que alguna cosa no funciona como estaba previsto, primero las letras que se encuentran en el extremo derecho de cada línea. Si no coinciden con las del libro, posiblemente ya habrá detectado el error.

Por otra parte debería emplear sólo los números de línea indicados para que mantener la capacidad de

funcionamiento de su programa.

Esto es importante, puesto que en el ulterior desarrollo se intercambiarán algunas líneas de programa y se introducirán algunas rutinas adicionales.

Se ruega además, que los propietarios de ATARI 600 XL tengan en cuenta las referencias del apéndice.

LA IDEA DEL JUEGO

Hemos intentado reducir al mínimo la teoría, y todo lo que falta es una historia que deseamos realizar como juego de aventuras. Entonces ya podemos conectar nuestro ATARI y comenzar con la práctica. Quizás usted tenga una idea determinada sobre su primera aventura, si no, puede inspirarse en las propuestas siguientes.

1. LA CASA FANTASMA

Usted recibe la última carta de su tía, donde se especifica que le deja en herencia su vieja residencia señorial. Su tía le llama la atención en su comportamiento, para que no le ocurra lo mismo que a su tío. Sin embargo, observaciones de esta clase no le retienen y usted va a ver la casa. Allí descubre una bibliioteca secreta con libros sobre conjuros, y un lugar de ofrendas en el sótano. Numerosos fantasmas le hacen la vida imposible, hasta que al fin se entera, que uno de sus antepasados había asesinado al líder de la secta para aumentar su própio poder, y que por ello recibió el castigo de ser enterrado vivo en un muro. Ahora, usted le quiere proporcionar el descanso eterno.

2. EL LEGADO DEL VIEJO

Su amigo y vecino, un famoso científico, ha perdido la vida en circunstancias misteriosas. Usted recibe una noticia suya, en la cual se le ruega que termine la obra de su vida. Después de conseguir entrar en su laboratorio hasta ahora secreto, su misión consiste en poner en marcha su superordenador. Este le dará entonces, más instrucciones.

3. ODISEA ESTELAR

Su nave espacial sufre tantos desperfectos por una tormenta cósmica, que se ve obligado a aterrizar en una estación sin tripulación, para conseguirse piezas de recambio y poder volver a la Tierra.

4. FIEBRE DE ORO

Usted tropieza en la selva con un hombre viejo y herido de muerte. Este le informa sobre su mina de oro. Le muestra algunas piezas de oro y le explica, que ha escondido oro en otros siete lugares deiferentes de la mina.El ya no lo necesita y da autorización para ir a buscar el oro.

Quisiera tomar esta última propuesta, una aventura del tipo busca del tesoro, como base para el próximo capítulo de este libro, y desarrollarla junto con usted. Naturalmente, usted es libre de realizar una idea propia, sin embargo, siempre volveré a hacer referencia a "Fiebre del oro" en todas las ampliaciones que le presentaré de aquí en adelante.

LA FORMACION DEL MUNDO

Con la elección del tema, el recorrido de la aventura ya está definido en términos generales. Ahora vamos a desarrollar algunas estructuras aisladas y las refinaremos cada vez más hasta llegar a crear un mundo totalmente previsto.

Procederemos de forma idéntica en la programación, construiremos paso a paso nuestra aventura y nos aseguraremos cada vez del correcto funcionamiento de cada una de las rutinas.

EL MAPA

Al principio soló existe la nada. A partir de aquí crearemos un mundo idoneo para nuestro juego. El tema del juego determina lo que nuestro mundo deberá ofrecer. La fiebre del oro tendrá lugar probablemente en unas minas.

Imaginemonos esta mina: un despeñadero escabroso lleno de rocalla, un cobertizo con herramientas, canales de madera que conducen el agua necesaria para el lavado de oro desde un pequeño arroyo de la montaña, troncos de madera quebradizos y secos que rodean la entrada sombría y pasadizos oscuros llenos de peligros.

Seguramente se nos ocurriran fácilmente numerosas otras imágenes por el estilo, que nos permitirán encontrar una cantidad suficiente de lugares para las acciones previstas.

Quisiera limitar este ejemplo concreto sólo a seis lugares de momento. Empezaremos con una excursión a través

de un bosque hasta que lleguemos a una mina, donde nuestra misión consiste en buscar los tesoros.

Con esto ya hemos fijado el comienzo provisional de la fiebre del oro. Pero para poder decidir sobre el éxito o el fracaso del jugador, también debemos fijar un final victorioso del juego, además de definir naturalmente varias situaciones perjudiciales durante el desarrollo de nuestro proyecto.

Así podemos contentarnos con la búsqueda de todos los tesoros, aunque también podemos plantear al jugador la misión de depositar en un lugar seguro el tesoro hallado. Pero de momento, nos limitaremos a encontrar el oro.

Ya hemos fijado misión y objetivo, así como dos lugares esenciales de la acción. Por un lado tenemos el bosque, donde se inicia la aventura, construido a partir de dos lugares, y por otro lado tenemos la mina. De todas maneras sabemos perfectamente, que no podemos crear ninguna acción en este reducido mundo. Necesitamos sitio para poder esconder algunos objetos y tenderle algunas trampas al jugador.

Tampoco es normal, que unas minas empiecen entre las raíces de unos árboles. Debemos crear transiciones adecuadas del paisaje, para que el juego tenga una configuración real.

Con tres lugares más disponemos de suficiente sitio para el desarrollo de la primera versión de nuestra aventura. Ahora podemos formular una lista de todos los lugares que se presentan:

en el bosque
en el bosque
linde del bosque limitado por un despeñadero
un claro del bosque

claro del bosque en una vertiente
Entrada a las minas

El siguiente paso hacia el programa terminado, es la elaboración conveniente de un mapa en el que se representen todos los lugares en forma de rectángulos. Estos rectángulos estarán numerados y provistos de su correspondiente descripción, donde ya nos preocuparemos por una formulación adecuada y donde consideraremos en lo posible la longitud de línea de 38 caracteres que querremos utilizar con nuestro Atari. Así, nuestra descripción debe poder añadirse sin problemas al texto preliminar "Estoy".

El número que recibe cada lugar no tiene ninguna importancia. Debemos empezar por el uno y continuar contando en orden creciente, con lo cual también evitamos la doble numeración. A continuación conectamos los lugares entre si y anotamos el punto cardinal correspondiente en cada límite.

En este mapa podremos leer ahora sin más problemas, si el jugador se encuentra, por ejemplo, en el lugar 3, que se provocará emitir la siguiente informacion en la pantalla:

Estoy en el bosque. Delante de mi se alza un despeñadero abrupto.

Además, nuestro mapa nos da información sobre las posibles direcciones en que podremos movernos:

Puedo ir hacia Oeste, Sur.

Ahora podríamos programar para cada lugar de nuestro juego una condición del tipo ''Si jugador en lugar 3, imprime 'Estoy en el bosque. Delante de mi se alza un despeñadero abrupto' '', pero con ésto llenaríamos en poco tiempo la

memoria disponible.

Sin embargo, el principio es correcto Memorizaremos todos los lugares en una lista, y como acceder al elemento requerido en cada caso para la emisión de la descripción.

Primeramente, indicamos las descripciones de lugar que deberán ser emitidas por pantalla:

```
200 REM ******** DESCRIPCIONES DE LUGAR
201 PRINT "EN EL BOSQUE."
202 PRINT "EN EL BOSQUE."
203 PRINT "EN EL BOSQUE,  DELANTE DE UN
DESPEÑADERO."
204 PRINT "EN UN CLARO DEL BOSQUE."
205 PRINT "EN UN CLARO AL LIMITE DE UNA
VERTIENTE."
206 PRINT "EN LA BOCA DE ENTRADA DE UNA
VIEJA MINA."
```

Tal como muestra la secuencia de un programa, los lugares ya constan en la memoria de nuestro ordenador. Ahora debemos encontrar una manera de acceder a ellos directamente.

Para ello nos servimos de la técnica de la subprogramación, que se ofrece especialmente con los ordenadores Atari por sus posibilidades de saltos condicionados. Vayamos a elaborar una subrutina para cada uno de los lugares, que se encarga solamente de la emisión de sus descripciones.

Un subprograma se llama normalmente, mediante GOSUB y se finaliza con RETURN. Ampliemos pués, las líneas 201 a 206, encargadas de los seis primeros lugares:

```
200 REM *********DESCRIPCIONES DE LUGAR
201 PRINT "EN EL BOSQUE.":RETURN
202 PRINT "EN EL BOSQUE.":RETURN
203 PRINT "EN EL BOSQUE DELANTE  DE  UN
DESPEÑADERO.":RETURN
204 PRINT "EN UN CLARO DEL BOSQUE.": RE
TURN.
205 PRINT "EN UN CLARO AL LIMITE DE UNA
VERTIENTE.":RETURN
```

```
206 PRINT "EN LA BOCA DE ENTRADA DE UNA
MINA.":RETURN
```

La elección del lugar, sólo depende, naturalmente, del lugar donde se encuentre el jugador. El destino del salto corresponde por lo tanto, a la línea 200 más el número del lugar actual.

Supongamos, que el jugador está en el lugar 3 (jugador=3), entonces, correspondiendo a la entrada de GOSUB 200+jugador, el Atari imprimirá la descripción correcta.

Tras la entrada de las líneas siguientes, ya seremos capaces de inspeccionar cada lugar:

```
190 GOTO 1390

1080 PRINT
1110 PRINT "ESTOY ";:GOSUB 200+JUGADOR
1390 PRINT "JUGADOR EN QUE LUGAR ";:
INPUT JUGADOR
5000 GOTO 1080
```

Con ello se permite investigar nuestro mundo, pero la forma de continuar el movimiento no es satisfactoria. No queremos saltar confusamente de un lugar a otro, si no que pretendemos orientarnos de manera concreta con ayuda de los puntos cardinales.

Normalmente, el lugar que el jugador acaba de pisar, suele tener un mínimo de una y un máximo de seis salidas. Al lugar 1 limita en el Este el lugar 2; en las direcciones Norte, Sur y Oeste, así como hacia arriba y hacia abajo no conduce ningún camino.

Si acordamos que las direcciones se indicaran siempre en el mismo orden N, S, O, E, arriba y abajo, el lugar 1 puede especificarse de la manera siguiente:

```
1 en el bosque    -,-,-,X,-,-
```


Para ser más prácticos, no se marcan las salidas con una X, sino que se indica explícitamente el lugar al que se puede acceder en esta dirección:

```
1 en el bosque    0,0,0,2,0,0
2 en el bosque    0,0,1,3,0,0
```

Estos seis valores para cada lugar se memorizan en una variable, llamémosla PASADA. Como en este caso se señala el lugar de entrada mediante dos valores distintos (lugar y dirección), es conveniente efectuar la memorización en un campo bidimensional.

DIGRESION: VARIABLES & CAMPOS

Para almacenar resultados intermedios o datos de un programa, en todo tipo de lenguaje de ordenador se utilizan las variables.

Estas variables se comportan como pequeñas casillas, en las que se puede depositar algo para su conservación. Estas casillas tienen nombres, como pueden ser maletero o cajón de escritorio. Existen además casillas del mismo tipo: en el escritorio, uno debajo de otro, cajón 1, cajón, 2, 3, etc. Cada casilla se llama cajón y sólo se distinguen por su número. Lo mismo ocurre en BASIC: Podemos dar un nombre distinto a cada variable, o asignar un sólo nombre y un número de identificación a las del mismo tipo, colocando el numero entre parentesis detras del nombre de la variable.

Podemos distinguir entre dos tipos esenciales de variables. Unas solo admiten números, números con los que también se puede operar; el otro tipo sirve para admitir textos. La configuración y la longitud de los textos no tiene importancia, mejor dicho: permitiremos cualquier carácter del teclado y longitudes de hasta 32767 caracteres seguidos. Pero no debemos olvidar que hay que comunicar la longitud deseada al interpretador, lo cual puede efectuarse con una

instrucción DIM de la forma DIM TEXT$(longitud de cadena de carácteres).

Algunos ejemplos de variables correctas serían: A, A1, A2, TEXT1 (sólo para números), TEXT$ (para textos), como variables simples; A(1), A(2), A(3) como variables indexadas.

Hay que destacar aquí, que el BASIC del Atari no admite variables de cadena (strings) indexadas.

Quizás también haya observado, como algunas personas intentan decorar sus viviendas con la colocación de cajas de imprenta -que se crearon para guardar las letras de imprentas- repletas de miniaturas. Como principiantes que somos, imaginemos ahora cada una de estas casillas como una variable.

La situación de cada casilla se determina exactamente mediante sus coordenadas fila y columna. Esto significa que si queremos conocer el contenido de la casilla (2,3), miramos la casilla que se encuentra en la tercera fila de la segunda columna.

Esta disposición se llama campo bidimensional en el idioma de ordenador, o también array.

En una array como éste guardamos entonces, la información sobre cada lugar, y a cada lugar se le asigna una fila propia. O sea que necesitamos seis columnas para implementar la llamada travel - table o tabla de direcciones:

LUGAR	N	S	O	E	AR	AB
1	0	0	0	2	0	0
2	0	0	1	3	0	0
3	0	4	2	0	0	0
4	3	0	5	0	0	0
5	0	0	6	4	0	0
6	0	0	0	5	0	0


```
501 PASADA(1,1)=0:PASADA(1,2)=0
502 PASADA(1,3)=0:PASADA(1,4)=0
503 PASADA(1,5)=0:PASADA(1,6)=0
```

Las líneas anteriores programan la pasada del lugar 1 haçia el lugar 2 en dirección Este. De la misma manera podríamos proceder con los restantes lugares, pero derrocharíamos mucha memoria con ello (¡cuántas veces escribimos 'pasada' ?), además de que no resulta agradable tener que teclear tanto.

Por esta razón renunciamos a las líneas 501 a 503, y preferimos recordar en sù lugar las instrucciones DATA y READ.

INCISO: READ DATA

Normalmente utilizamos la instrucción INPUT para transferir datos a través del teclado a un programa que se está ejecutando. Tiene la forma INPUT Var, donde Var puede ser una variable cualquiera. Para la entrada secuencial de varios datos, se permite enlazar una cantidad suficiente de variables, separadas por coma, o sea, que INPUT LONGITUD, ANCHURA, ALTURA puede admitir tres números uno tras otro para su posterior uso en el programa.

Si los datos ya son conocidos al iniciar el programa, podrán ser incluidos directamente en el programa, hecho que sólo tiene sentido si los datos no varían para cada ejecución del mismo. Las llaves necesarias para ello son READ y DATA. INPUT es sustituido por READ: por esto, READ LONGITUD, ANCHURA, ALTURA cumple la misma función; los datos de entrada se preparan con DATA 1,2,2 en una línea cualquiera del programa.

Es importante saber que los datos serán leídos en el orden

en que están colocados. Esto significa, que el número de datos en líneas DATA debe ser igual que el número de variables de la instrucción READ, de lo contrario aparecerá un mensaje de error o no se consideraran todos los datos.

Si se desea volver a leer los datos empezando por el primer elemento, debe utilizarse la instrucción RESTORE.

A continuación, entramos las siguientes líneas en nuestro Atari:

```
150 DIM PASADA(AR,6)

500 REM              TABLA DE DIRECCIONES
501 DATA 0,0,0,2,0,0
502 DATA 0,0,1,3,0,0
503 DATA 0,4,2,0,0,0
504 DATA 3,0,5,0,0,0
505 DATA 0,0,6,4,0,0
506 DATA 0,0,0,5,0,0
```

Ahora deben transferirse los datos a las variables de trabajo. Para ello leemos primero el destino de las líneas DATA y lo escribimos luego en la correspondiente variable del campo PASADA. Para evitar errores, debe conocerse con antelación el número de lugares; así pues, introduciremos primero la variable AR = número lugares. Lo mas práctico sería inicializarla ya al principio del texto del programa, para que más adelante sea fácil efectuar ampliaciones de la aventura.

```
110 AR=6
190 GOTO 940
```


Ahora repasaremos dentro de un bucle todos los lugares:

```
940 FOR LUGAR=1 TO AR
```

e inicializamos la tabla de direcciones. Como deben leerse seis datos de direcciones para cada lugar, introducimos un segundo bucle dentro del primero:

```
950 FOR DIRECCION=1 TO 6
960 READ META : PASADA(LUGAR,DIRECCION)
=META
970 NEXT DIRECCION
980 NEXT LUGAR
```

Cada fila de nuestro array se identifica ahora con un lugar; en las seis columnas de cada fila se almacena el lugar que el jugador podrá pisar mediante introducción de la correspondiente dirección.

Esta disposición nos facilitará considerablemente la programación de los movimientos que deberá realizarse a continuación.

Antes de que este pueda moverse hacia algún lugar, se ha de informar sobre las salidas de que dispone. Tenemos que desarrollar algunas líneas de programa, que emitan en la pantalla los puntos cardinales disponibles.

Supongamos que nuestro personaje principal se encuentra momentáneamente en el lugar 3. Entonces nos dirigimos a la fila 3 de la tabla de direcciones:

LUGAR	N	S	O	E	AR	AB
1	0	0	0	2	0	0
2	0	0	1	3	0	0
3	0	4	2	0	0	0
4	3	0	5	0	0	0
5	0	0	6	4	0	0
6	0	0	0	5	0	0

Todo lo que debemos hacer es indicar el nombre de aquellas columnas que no contengan un 0. En nuestro ejemplo, son las columnas dos y tres, o sea, Sur y Oeste.

Como hemos visto, la fila del campo coincide con el lugar a investigar; para la preparación de este actual número de lugar, introdujimos antes la variable JUGADOR. Así que para emitir las direcciones que no están cerradas, el programa debe comprobar las seis direcciones del lugar JUGADOR y mostrar los nombres de todas las columnas cuyo valor sea distinto de cero.

Hay que comprobar seis direcciones:

```
1230 FOR DIRECCION=1 TO 6
1240 IF PASADA(JUGADOR,DIRECCION)<>0
THEN PRINT "*** SALIDA ***"
1280 NEXT DIRECCION
```

Al comprobar la ejecución, el programa describirá ahora cada lugar, así como cada posible salida; aunque desgraciadamente aun no indica su dirección.

Por suerte, para comprobar si existe una salida, siempre se dispone del número de la dirección recien examinada en la variable contadora DIRECCION, por lo que podremos proceder del mismo modo que para la emisión de las descripciones de lugar:

```
990 GOTO 1080

1010 REM ***** TEXTO CLARO: DIRECCIONES
1011 PRINT "NORTE":RETURN
1012 PRINT "SUR":RETURN
```



```
1013 PRINT "OESTE":RETURN
1014 PRINT "ESTE":RETURN
1015 PRINT "ARRIBA":RETURN
1016 PRINT "ABAJO":RETURN

1240 IF PASADA(JUGADOR,DIRECCION)<>0
THEN GOSUB 1010+DIRECCION
```

Iniciamos nuestro programa para efectuar un control, entrando algunos números de lugar, mirando al mapa podremos convencernos de que hasta ahora todo funcióna correctamente.

Después de estos preparativos, ahora haremos posible el movimiento dirigido en nuestra aventura.

Cualquiera que ya hubiera cargado alguna vez un juego de aventuras, sabrá que la introducción de un punto cardinal es la indicación más frecuente para la figura del juego. Por eso, nosotros como programadores, deberíamos facilitarle la labor al jugador, considerando como suficiente la entrada de la letra inicial, ahorrando de esta manera al aventurero el haber de teclear cada vez el nombre completo del punto cardinal.

Desgraciadamente, pueden encontrarse muchos juegos de aventuras en el mercado, que no ofrecen ninguna rutina independiente de la codificación y ejecución de las instrucciones y que exigen un explícito VE HACIA OESTE.

Pero nosotros queremos entrar nuestras indicaciones de dirección de forma abreviada, porque podremos tratarlos con preferencia, lo que favorece por su parte, a una reacción más rápida del programa a la entrada.

Sustituyamos, primero, la línea 1390, para permitir entradas correspondientes a la aventura:

```
150 DIM PASADA(AR,6),ENTRADA$(20)

1390 PRINT  "QUE DEBO HACER ";:INPUT EN
TRADA$
```

Antes de que nuestro programa ejecute ahora cualquier manipulación del juego, debería comprobar la longitud de la entrada del jugador. Si la longitud es superior a dos letras (arriba=AR), se tratará de alguna acción, si no, debe ejecutarse la rutina del movimiento. Esta comprueba en primer lugar, si el camino en la dirección señalada esta libre, y si lo está, se leerá el nuevo lugar en la deseada tabla de direcciones bajo la correspondiente dirección (columna) del lugar, en que se encuentra el jugador (fila), que se asignará a la variable correspondiente (JUGADOR). Para finalizar, el jugador será informado sobre la ejecución de la acción:

```
1080 PRINT
1400 IF LEN(ENTRADA$)>2 THEN 1480
1410 IF ENTRADA$="N" AND PASADA(JUGADOR
,1)<>0 THEN JUGADOR=PASADA(JUGADOR,1):
PRINT "O.K.":GOTO 1080
1420 IF ENTRADA$="S" AND PASADA(JUGADOR
,2)<>0 THEN JUGADOR=PASADA(JUGADOR,2):
PRINT "O.K.":GOTO 1080
1430 IF ENTRADA$="O" AND PASADA(JUGADOR
,3)<>0 THEN JUGADOR=PASADA(JUGADOR,3):
PRINT "O.K.":GOTO 1080
1440 IF ENTRADA$="E" AND PASADA(JUGADOR
,4)<>0 THEN JUGADOR=PASADA(JUGADOR,4):
PRINT "O.K.":GOTO 1080
1450 IF ENTRADA$="AR"  AND PASADA(JUGAD
OR,5)<>0  THEN JUGADOR=PASADA(JUGADOR,
5):PRINT "O.K.":GOTO 1080
1460 IF ENTRADA$="AB"  AND PASADA(JUGAD
OR,6)<>0  THEN JUGADOR=PASADA(JUGADOR,
```



```
1470 PRINT"HACIA ALLI NO CONDUCE NINGUN
CAMINO !":GOTO 1080
1480 REM A PARTIR DE ACCIONES POSTERIORES
2000 GOTO 1080
```

Naturalmente con el inicio de cada juego debemos informar al programa sobre el lugar del comienzo.

```
120 JUGADOR=1
```

Un corto paseo por nuestro mundo de aventuras nos convencerá rápidamente, de que a continuación tendremos que ocuparnos de configurar la imagen de pantalla si queremos evitar que después de una cuantas jugadas se presente un lio informativo a nuestros ojos.

FORMATEAR LA SALIDA

Para posibilitar una limpia configuración de imagen hay que borrar naturalmente la pantalla al comenzar el juego:

```
10 GRAPHICS 0
1080 PRINT
```

Naturalmente no tiene sentido imprimir esta línea vacía en una pantalla limpia. Se hará necesaria siempre después de ejecutar una instrucción, ya que nuestra entrada, así como indicaciones posiblemente efectuadas, deben desplazarse una línea en sentido ascendente. Este desplazamiento se consigue con ayuda de un PRINT en la última línea de pantalla, y es necesario para finalizar cada jugada.

Para efectuar nuestras entradas, necesitamos la línea inferior, o sea, tras emitir la información al jugador, debemos mover el cursor hacia ella.

En cuanto a su capacidad, el BASIC del Atari nos ofrece dos posibilidades. Por un lado están los comandos de control del cursor, que pueden desplazar la marca de escritura por uno número determinado de líneas hacia abajo. Pero desgraciadamente, este valor depende de la cantidad de información emitida, o sea, de la cantidad de objetos, y no está disponible sin más ni más durante el transcurso del juego.

Sin embargo, profundizaremos también en esta posibilidad, ya que a continuación utilizaremos uno de los códigos de control.

De manera simplificada, puede decirse que cada instrucción existe para el interpretador de BASIC en forma de número codificado, el llamado "token", y no en forma de secuencia de letras.

Podemos dirigirnos a todas las rutinas del interpretador, previstas para la gestión de la pantalla a través de este número con ayuda de la función CHR$, asignando a cada instrucción de control del cursor lo siguiente

```
CHR$(28)  cursor arriba
CHR$(29)  cursor abajo
CHR$(30)  cursor izquierda
CHR$(31)  cursor derecha
```

Esto significa que la instrucción PRINT CHR$(28) desplazaría el cursor una línea hacia arriba.

Otra solución mucho más adecuada es la instrucción POSITION; que situa el cursor en una posición exactamente definida. Se utiliza en la forma "POSITION X,Y", donde Y corresponde a la línea y X la posición de escritura dentro de una línea:

```
1390 POSITION 2,23 : PRINT  "QUE DEBO
HACER";:INPUT ENTRADA$
```


Con ello, nuestra entrada tendrá lugar en la línea 24 (la línea superior es la línea 0), y el RETURN final desplaza la pantalla una línea hacia arriba, por lo que no es necesario tomar otras medidas respecto a las informaciones de las reacciones para el jugador. Con el comienzo de la nueva jugada, la línea 1080 vuelve a hacer disponible la fila inferior para el próximo ciclo de entradas.

Esta modificación provoca que de momento se realicen todas las emisiones en la línea inferior de la pantalla, mientras que las descripciones de lugar deben iniciarse en cambio, en la línea superior:

```
1100 POSITION 2,0:PRINT "ESTOY ";:GOSUB
200+JUGADOR
```

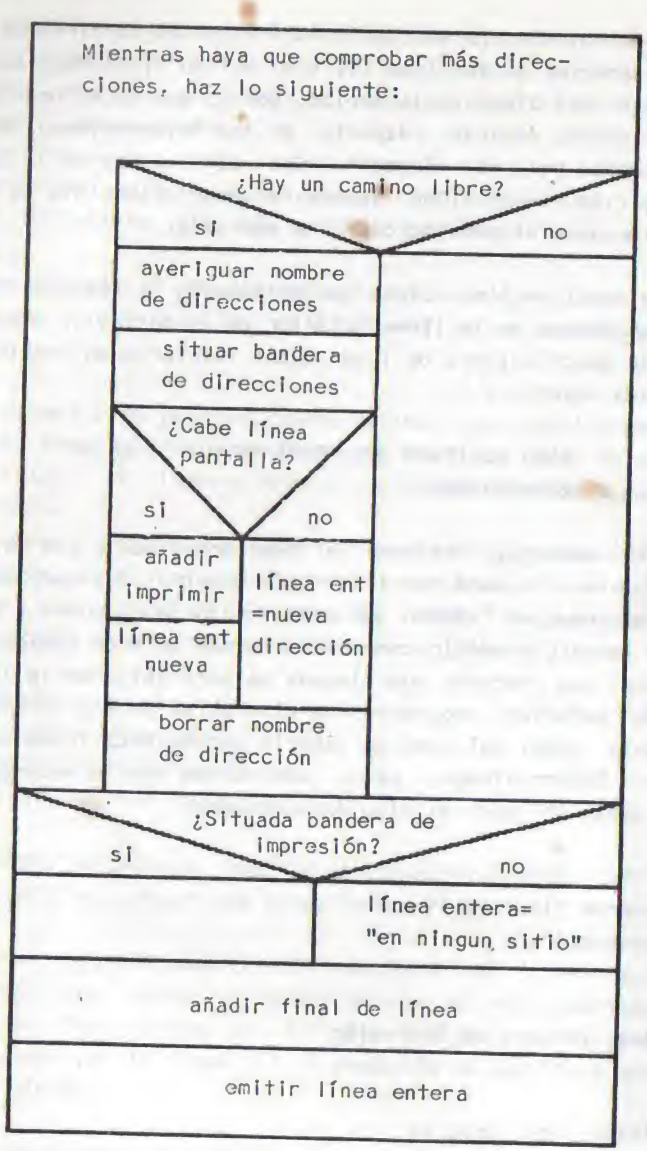
Sin embargo, informar al aventurero sobre las salidas disponibles, no será tan fácil para nosotros, ya que, aunque suprimiesemos el retorno del carro mediante el punto y coma, sería imposible emitir correctamente más de tres caminos.

Hay que procurar que ninguna palabra salga de la línea; además deberían separarse los diferentes puntos cardinales mediante coma, así como se debería cerrar cada frase con un punto. Desarrollemos, pues, una rutina que se encargue de esta cuestión según el siguiente esquema:

```
preparar final de línea:
backspace+"."


borrar bandera de impresión


iniciar línea entera:
"Puedo ir hacia"
```



Al confeccionar este diagrama Strukto se ha partido de las consideraciones siguientes:

PUEDO IR HACIA NORTE, SUR, OESTE, ARRIBA, ABAJO.

Está claro que antes de imprimir cualquier cosa, la línea de emisión debe comenzar con "Puedo ir hacia" (1220).

Ahora podrían imprimirse, de manera habitual, las direcciones en forma de texto descifrado dentro de un bucle. Desgraciadamente, al hacerlo no habría ninguna posibilidad de controlar la longitud de la emisión. Se ofreció como solución, crear primero una línea de impresión completa que cupiese justo en una línea de pantalla (1260, 1270).

Por ello se averigua cada texto, utilizando una variable auxiliar T$ (para el string parcial), y éste se sumara a la línea de impresión mientras no exceda la máxima longitud de línea. De esta manera pueden emitirse hasta tres indicaciones de dirección, mientras que una cuarta dirección ya superaría el valor límite de 38 caracteres.

Esto significa, que antes debe emitirse la línea de impresión hallada hasta ahora, y que la cuarta dirección se registre como nuevo contenido en DZ$.

Este procedimiento se repite con el tratamiento de la segunda línea de impresión, hasta que se hayan comprobado las seis direcciones.

La próxima misión consiste en emplazar un punto al final de la frase. La línea 1040 crea el string ZENDE$ (final de línea) para este fin, que mueve el cursor dos posiciones hacia atrás e imprime entonces un punto. Este string se añade al último texto a imprimir en DZ$; a continuación, se emite el contenido de esta variable.

Un último problema se ocasiona por la situación - que podría darse - de que el jugador pueda encontrarse en un lugar que

aparentemente no tiene salidas. En este caso, la rutina creada hasta ahora daría una imagen algo fea en la pantalla. Por esta razón se prevé otra variable auxiliar, que sirve como señal para saber si se ha imprimido al menos una dirección.

Para ello se aprovecha el hecho de que el ordenador considera siempre como verdadero un contenido de variable de -1, y cualquier otro como falso. Así, en el caso de que no se haya emitido nada (GEDRUCKT<>-1), puede añadirse el texto ''en ningún sitio'' a la línea de impresión.

Con ello se necesitan efectuar las siguientes modificaciones y ampliaciones:

```
150  DIM PASADA(AR,6), ENTRADA$(20),DZ$
(42),T$(40),ZENDE$(3)
1011 T$="NORTE, ":RETURN
1012 T$="SUR, ":RETURN
1013 T$="OESTE, ":RETURN
1014 T$="ESTE, ":RETURN
1015 T$="ARRIBA, ":RETURN
1016 T$="ABAJO, ":RETURN

1040 ZENDE$=CHR$(30):ZENDE$(LEN(ZENDE$)
+1)=CHR$(30):ZENDE$(LEN(ZENDE$)+1)="."
1220 DZ$="PUEDO IR HACIA ":GEDRUCKT=0
1240 IF PASADA(JUGADOR,DIRECCION)=0THEN
GOTO 1280
1250 GOSUB 1010+DIRECCION:IMPRIMIDO=-1
1260 IF LEN(DZ$)+LEN(T$)>=38 THEN PRINT
DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 1280
1270 DZ$(LEN(DZ$)+1)=TS
1280 T$="":NEXT DIRECCION
1290 IF NOT IMPRIMIDO THEN DZ$(LEN(DZ$+
1)="NINGUN SITIO."
1300 DZ$(LEN(DZ$)+1)=ZENDE$
```



```
1310 PRINT DZ$
```

ORO, PLATA Y OTRAS COSAS UTILES

Este paseo a través de nuestro mundo de aventuras ofrece al aventurero una imagen triste y poco interesante hasta ahora, puesto que no dispone de ningún tipo de objetos para la acción. Además, simples descripciones de lugar no permiten tampoco realizar ningún juego. Volvemos a coger pues nuestro mapa, y enriquecemos nuestro mundo con los objetos deseados, limitándonos a objetos que sean importantes para seguir el desarrollo del juego. Como simples adornos, el posterior trabajo sería excesivo, ya que debemos permitir que el jugador pueda coger con la mano y examinar todo lo que encuentre. De todas maneras, no podemos evitar situar como mínimo un objeto en cada lugar, ya que es muy dudoso que el aventurero no vea absolutamente nada sin más ni más. Si no disponemos de suficiente fantasía, el objeto puede ser tranquilamente "nada especial". Esta formulación se encuentra muy a menudo en los juegos de aventuras, ¿porqué? sirve para reducir el volumen de trabajo, ya que en un bosque los objetos como Oarboles, arbustos, hierba, insectos, etc. no son realmente nada especial.

Pero a menudo, no se puede evitar el uso de cosas inútiles, especialmente cuando forman parte del ambiente y la aventura debe ofrecer una impresión realista.

Así también nosotros tendremos que colocar árboles en nuestro bosque, aunque la acción no lo requiera. En el lugar dos, que ya se encuentra cerca de las montañas, algunos pedruscos refuerzan la impresión de realidad. Además, a una mina le corresponde una cabaña, no siendo importante si es la vivienda del propietario de la mina o si sirve para guardar herramientas. Finalmente, debemos esconder en algún

Lugar los tesoros que hay que buscar. Además es conveniente introducir algunos objetos que perjudiquen al jugador .

Permítame pues, centrarme en el desarrollo de la acción y escoger los objetos adecuados.

DESARROLLO DE LA ACCION FIEBRE DEL ORO

La versión 1 debe encargar al jugador la misión de encontrar dos tesoros, monedas de oro y de plata en los alrededores de la mina. Al comenzar el juego, el jugador se encuentra en el bosque y debe acercarse a las minas. En su camino hacia ellas, en el bosque descubre un hoyo, en cuyo fondo se encuentra una pesada arca de hierro. En ella se halla la plata, pero desgraciadamente, la arca esta cerrada con un candado y una gruesa cadena de hierro; en ningún sitio se puede encontrar una llave que funcione. Sin embargo, en la boca de entrada de la mina hay unas barras de hierro que son lo bastante fuertes como para partir con ellas la cadena de hierro o romper el candado del arca. También podemos permitir que el jugador intente conseguir su objetivo con ayuda de un explosivo; dejemos que encuentre dinamita en una cabaña de madera, una buena ocasión para nosotros de programar su final anticipado. Colocaremos las piezas de oro en una cueva que, muy a pesar del jugador, está habitada por un oso feroz. Pero de esto no se enterará el jugador hasta que inspeccione la cueva. Si le vence la codicia por el metal amarillo y le induce a cogerlo espontáneamente, tambien el hambre del oso hara

que este pierda el miedo a los hombres - después de la dinamita, está es la segunda trampa para el jugador.

Por otra parte, gracias a Wilhelm Busch (autor de poesías infantiles), todos sabemos que a los osos les encanta la miel. Por esta razón, en una estantería de la cabaña colocamos una botella que contenga este codiciado líquido. Si el jugador entra en la cueva con esta botella, el oso olerá la miel y desaparecerá con la botella en las profundidades de la cueva, con lo que ya no habrá nada que se oponga al final feliz de esta miniaventura.

El mapa de nuestro juego de aventuras "Fiebre del oro" se parecerá ahora al siguiente esquema.

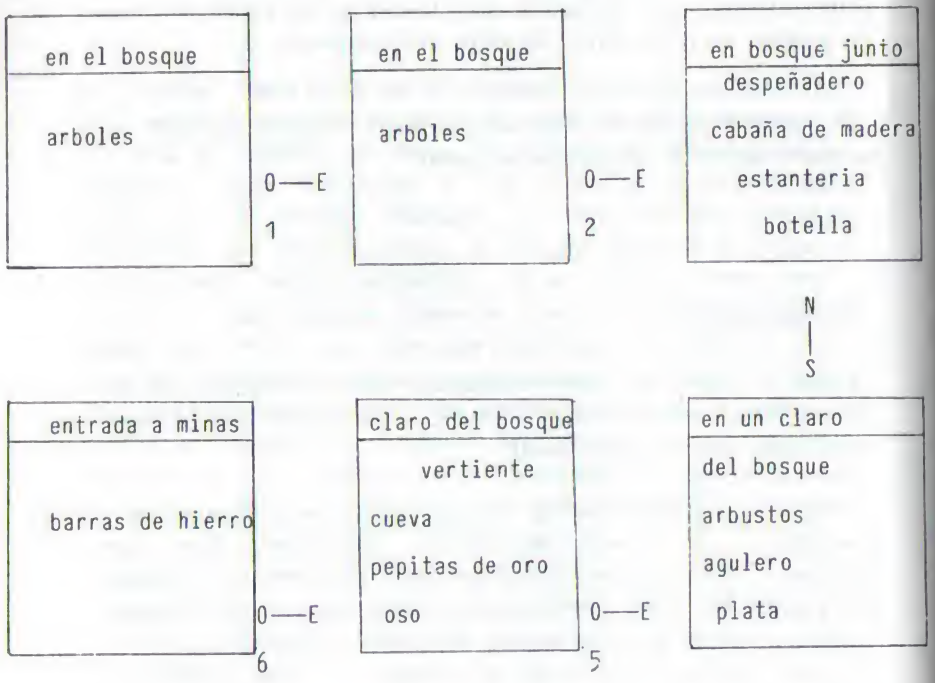
LOS OBJETOS

Antes de completar nuestra aventura con la programación de los objetos necesarios, tendremos que hacer aun algunas reflexiones preliminares aún;

Para conseguir que el jugador entre en este mundo del ordenador, no debemos describirlo sécamente, sino que tendremos que esforzarnos en mantener despierto el interés del jugador mediante descripciones plásticas. La información ´ Veo un oso.´ permite seguir el juego al jugador, pero la noticia ´ Veo un oso con una mirada muy feroz´le daré la ocasión de vivirla.

Si en el aventurero pensase en hacer un nuevo amigo, sería fácil para nosotros programar una entrada como ´Acaricia oso con mirada muy feroz´ pero no se le podría exigir tal tarea al jugador.

Para evitar eso, tendremos que llamar a cada objeto por un



nombre, además de ofrecerle la propia descripción del objeto, con lo que también obtenemos la posibilidad de fijar libremente la longitud de palabra de la aventura según nuestras necesidades. Probablemente usted se haya dado cuenta de que las aventuras profesionales, procedentes de lugares de habla inglesa, suelen requerir sólamente las tres o cuatro primeras letras para identificar cada palabra. Normalmente bastan también tres letras para definir los términos de las aventuras castellanas, pero al realizar un proyecto más amplio se aconseja confeccionar antes una lista de los objetos previstos y repasarla atentamente. De lo contrario, al proceder a la programación de las condiciones y acciones, se sorprenderá de la cantidad de sustantivos importantes para la aventura que empiezan con las mismas tres letras. Si su juego de aventuras se compone además de letras mayusculas y minusculas, deberá tener presente utilizar solamente letras mayusculas para definir los nombres, ya que estas abreviaciones serán comparadas más adelante con las entradas efectuadas por el jugador.

Como tercera información, además del nombre completo y del abreviado, tendremos que memorizar también el número del lugar en el que se encuentra el objeto. Estos números se modifican continuamente durante el juego y no pueden ser leídos de una línea de programa fija, sino que deben estar disponibles en determinadas variables.

Realizamos, pues, una lista en la que cada valor corresponde al paradero de un objeto.

Por razones prácticas utilizamos la variable indexada OB(), que nos ofrece una manera simple de controlar y manipular la posición de cualquier objeto.

Mientras que un objeto no participe activamente en el juego, el valor de OB = 0, lo que equivale a derir, que el objeto se encuentra en el almacén. En otros casos, el contenido de OB corresponde al valor de la variable jugador, siendo visible en el lugar, o bien su valor es -1, en cuyo caso el

aventurero lo lleva consigo (Inventory).

Las próximas líneas preparan las descripciones de todos los objetos, necesarios para la realización del programa confeccionado hasta ahora (fiebre del oro, miniversión), para la salida en pantalla.

Cada línea correspondè a un pequeño subprograma, puesto que queremos utilizar la técnica ya desarrollada con los puntos cardinales.

```
300 REM ********************** OBJETOS
301 T$="MUCHOS ARBOLES GRANDES":RETURN
302 T$="MUCHOS ARBOLES GRANDES":RETURN
303 T$="ALGUNAS ROCAS":RETURN
304 T$="UNA CABAÑA DE MADERA EN RUINAS
" :RETURN
305 T$="UNA SUCIA BOTELLA FORRADA":RETU
RN
306 T$="MIEL":RETURN
307 T$="UNA CAJA DE MADERA":RETURN
308 T$."UNA ESTANTERIA DESVENCIJADA":RE
TURN
309 T$="UN POCO DE EXPLOSIVO":RETURN
310 T$="UN AGUJERO OSCURO":RETURN
311 T$="UN ARCA DE HIERRO OXIDADA":RETU
RN
312 T$="*MONEDAS DE PLATA*":RETURN
313 T$="UNA CUEVA OSCURA":RETURN
314 T$="UN OSO RABIOSO":RETURN
315 T$="ARBUSTOS BAJOS":RETURN
316 T$="VARIAS BARRAS DE HIERRO":RETURN
317 T$="*PEPITAS DE ORO*":RETURN
```


A cada uno de estos objetos le asignaremos ahora una posición dentro de nuestro mundo de aventuras:

```
400 REM ************ POSICIONES OBJETOS
410 DATA 1,2,2,3,0,0,3,0,0,0,0,0,5,0,4,
6,5

900 FOR OBJETO=1 TO AO
910 READ SITUACION:OB(OBJETO)=SITUACION
920 NEXT OBJETO
```

Ahora visualicemos las instalaciones de un lugar al jugador.

En la variable JUGADOR encontramos el número del lugar en que se ha entrado, OB() contiene el número del lugar de todos los objetos. Comprobemos pues dentro de un bucle, si el número de posición de un objeto coincide con el actual número de lugar:

```
1110 DZ$="VEO ":IMPRIMIDO=0
1120 FOR I=1 TO AO
1130 IF OB(I)<>JUGADOR THEN GOTO 1170
1140 IMPRIMIDO=-1:GOSUB 300+I:T$(LEN(T$
)+1)=", "
1150 IF LEN(DZ$)+LEN(T$)>=38 THEN PRINT
DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 1170
1160 DZ$(LEN(DZ$)+1)=T$:T$=""
1170 NEXT I
1180 IF NOT IMPRIMIDO THEN DZ$(LEN(DZ$)
+1)="NADA ESPECIAL. "
1190 DZ$(LEN(DZ$)+1)=ZENDE .
1200 PRINT DZ$
```

La línea 1110 del programa posterga la bandera de señal, que

más tarde necesitaremos para poder comprobar si se ha imprimido alguna cosa, y fija el comienzo de la línea de impresión.

La 1130 comprueba para cada objeto, si este se encuentra en un lugar diferente que el jugador. Si este test resulta positivo, se prosigue inmediatamente a comprobar el objeto siguiente; en caso contrario, se coloca la bandera de impresión, se reclama la descripción del objeto y se le añade una coma. Este proceso se repite hasta conseguir una longitud completa de línea de impresión de 38 caracteres y emitir el texto, o bien hasta que se haya ejecutado todo el bucle y emitido en consecuencia, todos los objetos visibles.

Finalmente, la línea 1190 reemplaza el último caracter y pone fin a la rutina de impresión con un punto.

Para llegar a conseguir la estructura de pantalla de las aventuras originales americanas, ya sólo nos falta añadir una línea de separación y una línea vacía:

```
150 DIM ....., BLANCOS$(38)

1020 BLANCOS$="
"
1210 PRINT BLANCOS$
1330 PRINT"----------------------------
---------":REM 37 X CONTROL R
```

Por otra parte, después de haber efectuado varias entradas, nos daremos cuenta de un último problema. Las informaciones que se encuentran en la mitad inferior de la pantalla se van desplazando cada vez más hacia arriba, mezclandose después de unas cuantas jugadas, con las descripciones de las inmediaciones del lugar. Por ello, el programa tiene la

Misión de borrar las líneas superiores antes de emitir estas descripciones, lo que puede conseguirse mediante la impresión de suficientes líneas vacías.

```
1090 FOR LINEA=0 TO 10:POSICION 2,LINEA
:PRINT BLANCOS$:NEXT LINEA
```

EL LEXICO

Antes de proceder ahora a programar el propio juego, y de ocuparnos de las acciones del programa invisibles para el jugador, debemos entrar los verbos necesarios. Junto con los mencionados nombres de los objetos, estos verbos formarán el léxico completo de nuestra aventura, siendo por ello la base de comunicación con el jugador.

Estas son las palabras que tendremos que comparar con las entradas efectuadas por el aventurero, o sea, que se van efectuando continuamente comprobaciones de su identidad. Sin embargo, llevaría demasiado trabajo programar, de manera parecida al tratamiento de los objetos, una línea propia para cada término. Ahora aprovecharemos nuestro acuerdo de elegir una longitud de palabra definida exactamente, con lo que conocemos perfectamente el número de carácteres significativos de cualquier entrada. Para ello es práctico analizar partes de igual longitud de un string largo. Necesitaremos dos de estos strings, uno para los verbos y otro para la descripción abreviada de los objetos:

```
160 DIM VERBOS$(40), OBJETOS$(80)
170 VERBOS$="INSPCOGEPON ABREUTILDEST"
180 OBJETOS$="ARBOARBOROCACABABOTEMIELC
AJAESTAEXPLAGUJARCAPLATCUEVOSO ARBUBAR
RNUGG"
```

ANALISIS DE LAS ENTRADAS

Ahora ya se reconocen y se ejecutan correctamente los cambios de direcciones efectuados por el jugador, un grupo de instrucciones con longitudes de palabra de una o dos letras. Sin embargo, la entrada estandar siempre consta de verbo y objeto, donde las longitudes de los términos no están sujetos a ninguna restricción, prescindiendo de la longitud mínima exigida para reconocer los conceptos. Para poder reaccionar tal como desea el jugador, nuestro programa debe comprobar su entrada:

¿está programado el verbo utilizado ?
¿está previsto el objeto ?

Si ambas preguntas pueden contestarse afirmativamente, la entrada se descompone en verbo y objeto, y se averiguan sus respectivos números. Si las componentes de la entrada no estan definidas, se emitirá un mensaje consecuente para el aventurero.

INCISO: TRATAMIENTO DE STRINGS

La principal diferencia entre un ordenador y una calculadora de bolsillo programable, es la capacidad del ordenador para procesar textos. Naturalmente, una calculadora no comprende los textos, sino que los entiende como una colección de determinados caracteres, la llamada variable de cadena o string. Estos string son, en general, combinaciones de todos los caracteres que se pueden introducir a través del teclado. El sistema operativo del ordenador ofrece ahora una serie de funciones para tratar estas variables de cadena.

Además de una serie de instrucciones de conversión como VAL() y STR$(), que convierten un número en un string y viceversa, nuestro Atari ofrece la posibilidad de concatenar

Y de descomponer strings. Con la instrucción B$=A$(1,3) podemos preparar las tres primeras letras de la variable de cadena A$ en B$, para disponer de ellas en su posterior tratamiento dentro del programa. En el caso de A$="ABCDEF", el contenido de B$ sería, por ejemplo, "ABC".

Para poder crear ahora este tipo de substring al efectuar la llamada de esta función, deben conocerse como parámetros los valores inicial y final. Esto puede hacerse a través de un acuerdo (en nuestro string OBJETOS$, después de cada tercera letra comienza una nueva palabra), así como mediante la búsqueda de una determinada señal dentro de un string.

Así por ejemplo, nuestro programa de aventuras busca la posición del espacio entre verbo y objeto de la entrada efectuada por un jugador y, utilizando este número de identificación, hallará los dos substrings EVERBO$ y EOBJETO$.

Sin embargo, para poder determinar el número de letras necesarias, debe hallarse previamente la longitud total de la entrada mediante LEN(ENTRADA$).

Si la ejecución de una entrada no se efectua en las líneas 1410 a 1470, se considerará, en el caso de que la longitud de la entrada conste de menos de tres letras, que se trata de un error de entrada. Después de emitir ''¡Hacia allí no conduce ningún camino!'', comenzará una nueva jugada.

Si se trata de una entrada más larga, la ejecución del programa continuará en la línea 2000:

```
160 DIM ... EVERBO$(20),EOBJETO$(20)
1470 IF LEN(ENTRADA$)<3 THEN PRINT "HAC
IA ALLI NO CONDUCE NINGUN CAMINO !":GO
TO 1080

2000 LN=LEN(ENTRADA$)
2010 FOR I=1 TO LN
```

```
2020 IF ENTRADA$(I,I)<>" " THEN NEXT I
2030 EVERBO$=ENTRADA((I,I)
2040 IF LEN(EVERBO$)=LN THEN GOTO 2090
2050 EOBJETO$=ENTRADA$(I+1,LN)
```

Después de determinar la longitud de la entrada completa, dentro de un bucle se comprueba cada letra, comenzando por la izquierda, para ver si es un espacio. Al encontrar el espacio, ya se conoce la longitud del verbo (del primero al último caracter recién comprobado) y el verbo puede ser asignado a la variable EVERBO$. La variable de cadena que comienza a partir de la siguiente posición (debe saltarse el espacio) corresponde pues, al objeto de la entrada.

Si se ha utilizado una instrucción de una única palabra (HELP), la longitud del verbo coincidirá con la longitud de la entrada. Puesto que no es necesario preparar ningún objeto para su posterior tratamiento, se bifurca directamente (línea 2040) para el análisis del verbo (línea 2090).

```
2090 FOR I=1 TO LEN(VERBOS$) STEP 4
2100 VN=VN+1
2110 IF VERBOS$(I,I+2)=EVERBO$ THEN2140
2120 NEXT I
```

Dentro de otro bucle se comparan siempre las cuatro siguientes letras de nuestro string VERBOS$, que contiene los nombres de todos los posibles verbos del juego, con el verbo de la entrada hallado previamente. Un contador tiene preparado el número del verbo recién comprobado, de manera que se puede salir del bucle cuando se produzca una identidad. Si se ha ejecutado todo el bucle sin encontrar ningún substring adecuado, significa que el programador no había previsto este verbo, lo cual se comunicará al jugador:

```
2130 PRINT "NO ENTIENDO EL VERBO !": GO
TO 1080
```


A continuación se hallará el número del objeto, utilizando el mismo principio:

```
2140 FOR I=1 TO LEN(OBJETOS$) STEP 4
2150 N=N+1
2160 IF OBJETOS$(I,I+2)=EOBJETOS$ THEN
GOTO 2200
2170 NEXT I
2180 PRINT "NO ENTIENDO EL OBJETO !":GO
TO 1080
```

Después de entrar estas líneas, hemos superado una buena parte de nuestro trabajo de desarrollo. Las rutinas creadas estructuran convenientemente la pantalla, y crean también la comprensión para las entradas que pueda efectuar el jugador.

Averiguan cual de los posibles verbos ha sido utilizado por el aventurero y con que objeto desea efectuar la acción, con lo que se posibilita una bifurcación del programa hacia una línea del mismo, prevista para ejecutar dicha acción. Precisamente por estas funciones centrales de control, esta parte del programa se llama motor.

RESUMEN: ESTRUCTURA DE LOS PROGRAMAS

Esta claro que nuestros programas de aventuras constan de tres partes:

1. Datos de la aventura
2. Motor de la aventura
3. Ejecución de las jugadas

1. Los datos constituyen la base de cualquier juego. Se transfieren en parte, a unas variables de trabajo, o bien, se mantienen disponibles en pequeños subprogramas para ser transmitidos al programa principal.

2. El motor se ocupa de controlar la completa ejecución del programa. Estructura la salida a pantalla, recoge las entradas del jugador, las valora e inicia la ejecución de las instrucciones.

El motor es independiente de la aventura en cuestión, y puede utilizarse invariablemente para cualquier aventura.

3. La tercera parte constituye la propia aventura. Si se cumplen todas las condiciones necesarias para una acción, esta será ejecutada.

Para que no pierda de vista el conjunto, vamos a resumir aquí la estructura de nuestra aventura, indicando también las rutinas que se crearán a continuación.

Así nos muestra el carácter universal de este borrador.

Esto significa que bastan algunas modificaciones para poder transformar la aventura de texto en otra gráfica. De la misma manera, gracias a la sistemática estructura uniforme con líneas de programa muy definidas para los diversos bloques de funciones, será posible desarrollar un generador de aventuras.

BORRADOR DE LA ESTRUCTURA: AVENTURA EN TEXTO

DATOS

0	100	IMAGEN TITULAR
100	150	DATOS DE CONTROL
150	160	RESERVAR MEMORIA
170		STRING TOTAL VERBOS
180		NOMBRES ABREVIADOS DE OBJETOS
200	300	DESCRIPCIONES DE LUGAR
300	400	DESCRIPCIONES DE OBJETOS
400	500	SITUACION DE OBJETOS
500	600	COMUNICACIONES DE LUGARES
600	700	INFORMACION ESTANDAR
700	800	SEGUNDO TITULO EVENTUAL
800	900	INSTRUCCIONES GENERALES DE JUEGO
900	1000	INICIALIZAR DATOS DE JUEGO

MOTOR DE LA AVENTURA

1000	1020	TABLA DE DIRECCIONES
1030	1040	VARIABLES DE CONTROL
1080	1320	GESTION DE PANTALLA
1321	1398	SUCESOS ESPECIALES
1390		ENTRADA DE LA JUGADA
1391	1399	TRAMPAS Y OBSTACULOS
1400	1500	MOVER PERSONAJE PRINCIPAL
1500	1600	BALANCE
1600	1700	SAVE GAME
1700	1800	LOAD GAME
1800	1950	HELP, VOCABULARIO, INSTRUCCIONES
1950	2000	INTERRUPCION DEL JUEGO
2000	2180	ANALISIS DE LA ENTRADA
2200		EJECUCION TABLA DE SALTOS

EJECUCION

5000	30000	MOVIMIENTOS

¿CUANDO FUNCIONA QUE?

Imaginemos ahora al jugador que pretende encontrar nuestros dos tesoros. Con una sola mirada se da cuenta de que está en el bosque, rodeado por árboles y roças. El sabe, como jugador experimentado de aventuras o lector del cápitulo anterior, que lo más importante puede encontrarse en las cosas más comunes y de menor apariencia.

¿Cómo reaccionará?
¿Qué entradas debemos prever?

Lo más probable será, que pruebe continuar con instrucciones como INSPECCIONA BOSQUE, INSPECCIONA ARBOL, o INSPECCIONA PEDRUSCO.

COGE ARBOL será una entrada menos probable; sin embargo, para no apoyar a los críticos de nuestros programas, debemos programar un mensaje del tipo ''No tengo tanta fuerza.''. Además, informaciones frecuentes que muestren, que una entrada ha sido entendida, animan al aventurero para continuar el juego.

Por otra parte, si el resultado de cada dos o tres entradas fuera el mensaje ''¡No entiendo lo que quieres decir!'', el estímulo para apagar el ordenador sería demasiado fuerte.

Por eso, en la práctica sólo proyectamos para los lugares 1 y 2 la emisión de varias informaciones para el jugador, misión rápidamente realizada por una instrucción PRINT, porque en estos lugares aun no hemos previsto ninguna acción, sino que sólo pretendiamos ampliar nuestro mundo. Pero una vez el jugador haya entrado en el lugar 3 y haya descubierto la cabaña de madera, también la inspeccionara y descubrirá la estantería con la botella forrada. Naturalmente, está debe aparecer junto con la estantería y la cabaña, en la descripción de la escena ('Veo ...'); esto significa que debemos modificar su paradero. Otro quebradero de cabeza puede provocarnos la entrada "Coge

botella'': ahora debe volver a desaparecer la botella recién aparecida. No puede volver a su viejo paradero, ya que deberá ser listada cuando nuestro jugador efectúe un BALANCE.

También es lógico que el jugador no pueda proceder a inspeccionar la botella en el lugar 1, ni tampoco en el 2. De todas formas, deberían rechazarse todas las entradas referentes a la botella, mientras ésta no se encuentre en inmediata proximidad del jugador.

Sin embargo, esto es cuestión de gustos, depende de cada programador. Se puede considerar, que la botella ya se encuentra en la estantería desde tiempos remotos, y lo que ocurre es que el jugador no la ha visto hasta ahora.

Pero si el jugador está obligado, por las numerosas sorpresas propias de una aventuras, a volver a empezar desde el principio, ya conoce la posición inicial de diversos objetos, y se le puede permitir cogerlos, sin que antes hayan sido listados en la información "Veo ...".

De hecho, con ello incluso se facilita la programación del proceso de coger, ya que hay una condición menos a comprobar. Pero, al igual que la mayoría de los productores de aventuras, nos limitaremos a considerar, que sólo se pueden coger los objetos visibles.

Así que la cabaña de madera sólo puede ser inspeccionada por nuestro personaje principal si se encuentra en la proximidad. Descubriremos la miel después de coger la botella forrada con nuestras manos y abrirla. El oso sólo se comportara pacíficamente, si está distraido comiéndose la miel.

Antes de ejecutar cualquier instrucción, debe comprobarse, si se cumplen todas estas condiciones. Incluso nuestro corto

programa de sólo seis lugares requerirá una gran cantidad de líneas de programa para ello. Si consideramos sólo los verbos introducidos hasta aquí en el programa, inspecciona, coge, pon, abre, utiliza, destruye, así como los diecisiete objetos, ya tenemos que programar acciones para 102 diferentes entradas del jugador. Si escribiésemos una tras otra todas estas líneas de programa, sin ningún borrador especial, sería fácil encontrar una explicación para los largos tiempos de ejecución del programa. Sin embargo, para mantenerlos relativamente cortos, dividiremos la parte del programa en varios bloques.

En principio, disponemos de dos posibilidades distintas. Por un lado podríamos reservar la cantidad necesaria de líneas de programa para cada lugar, y reservar en ellas unas cuantas instrucciones para cada posible acción del jugador. Esta técnica suele ser muy frecuente. Tiene la ventaja de ser muy rápida gracias a la ejecución directa de una instrucción, ya que esta no tiene que estar preparada previamente de alguna forma. Otra ventaja es, que el programa es muy claro que puede ampliarse o editarse con facilidad, sin necesidad de consultar ninguna lista. Si desea emplear alguna vez esta ejecución de un programa de aventuras, la siguiente realización de nuestro lugar 1 debe bastarle como estímulo. Tenga en cuenta, que se trata de un ejemplo que no tiene nada que ver con nuestro borrador. Vaya con cuidado pues, en no escribir sobre líneas de nuestro programa confeccionado hasta ahora.

```
100 PRINT "QUE DEBO HACER":INPUT ENTRA
DA$
200 REM DIVIDIR ENTRADA$ EN VERBO$ Y OB
JETO$ (COMO CONVENIDO)
300 ON LUGAR GOTO 1000, 2000, 3000, 400
0, 5000, 6000,
301 REM DEPENDIENDO DEL NUMERO DE LUGAR
SALTAR A LUGAR CORRESPONDIENTE
```



```
1000 GRAPHICS 0:              REM LUGAR 1
1010 PRINT "ESTOY EN EL BOSQUE".
1020 PRINT "VEO MUCHOS ARBOLES GRANDES"
1030:
1040 REM IMPRIMIR MAS OBJETOS
1100 PRINT "PUEDO IR HACIA SUR, ESTE."
1200 IF ENTRADA$="S" THEN LUGAR=6: GOTO
100
1210 IF ENTRADA$="E" THEN LUGAR 2: GOTO
100
1260 IFLEN(ENTRADA$)<3THENPRINT"HACIA A
LLI NO CONDUCE NINGUN CAMINO!":GOTO100
1300 IF ENTRADA$="INV" OR "INVENTARIO"T
HEN GOTO 200
1400 IF VE$="ABA" AND OB$="OSO"THEN PRI
NT "NO VEO NADA ESPECIAL": GOTO 100
1410 IF VE$="COG" AND OB$="OSO"THEN PRI
NT "NO TENGO TANTA FUERZA.": GOTO 100
1999 PRINT "NO ENTIENDO LO QUE QUIERES
DECIR>": GOTO 100
2000 a partir de aqui tratamiento del l
ugar 2
3000 a partir de aqui tratamiento del l
ugar 3
```

Si se ha fijado bien en las líneas precedentes, o si las ha entrado en su Atari, descubrirá que se puede confeccionar una aventura que funcione de manera aún más fácil, y se preguntará posiblemente, ¿por qué no hemos utilizado este borrador, que además puede entenderse sin necesidad de extensos comentarios ?

Bien, por un lado, las aventuras muy extensas requieren bastante más memoria, porque no podríamos evitar, ejecutar varias veces rutinas idénticas, ya que, para explicarlo

un ejemplo conocido, debemos ser capaces de depositar la botella en cada lugar (seis líneas de programa idénticas), y también volver a cogerla. Frente a estas doce líneas sólo existen dos lineas de programa en nuestro sistema de aventuras. Sólo este hecho justifica nuestra elección, pero también los argumentos de claridad y facilidad para editar pueden refutarse rápidamente.

Imaginese que para la ampliación de la aventura se necesiten más botellas para continuar la acción, de manera que preferimos convertir, por este o cualquier otro motivo, la botella en un pequeño barril de madera. Si en este momento su aventura se componía de 40 lugares, ahora podrá convertir un "Coge botella" por un "Coge barril" en las numerosas líneas correspondientes; lo mismo vale para "Pon botella", "Abre botella" etc.

Si por el contrario, ha seguido nuestro sistema de aventuras, basta con modificar tres letras dentro de nuestro string de la línea 180, y ¡¡listos!

Olvidemos, pues, este método y continuemos con nuestra aventura. También hacemos que la acción transcurra en líneas de programa exactamente definidas, pero cada bloque contiene ahora el tratamiento de un sólo verbo.

Nuestro motor ya ha averiguado los números de verbo y objeto, y por ello podemos dirigirnos directamente a estos bloques. Así es lógico, asignar líneas de programa a partir de 5000 a todas las líneas, que están ocupadas en ejecutar la instrucción "inspecciona". La acción "coge" se realiza correspondientmente a partir de línea 6000.

El motor realiza el control de la ejecución del programa con ayuda del número de verbo en la línea 2200:

```
2190 REM        INS   COG   PON   ABR
UTI   DES
```



```
2200 ON VN GOTO 5000,6000,7000,8000,900
0,10000
```

Con la linea de programa 5000 comienza ahora el bloque "INSPECCIONA". Es fácil comprender que ello requiere mucho trabajo.

No sólo debe emitirse la información ''EN LA CABAÑA HAY UNA ESTANTERIA VIEJA'' para contestar a la instrucción ''INSPECCIONA CABAÑA'', y aparecer la estantería como objeto visible. No, también debemos asegurar que la cabaña este disponible en el lugar visitado en este momento y transmitir, en caso necesario, un mensaje como "AQUI NO VEO NADA PARECIDO" al jugador.

Estas reflexiones demuestran el trabajo que lleva la programación de aventuras. Pero afortunadamente, también pueden contestarse numerosas entradas diversas del jugador con una sola línea de programa.

Así que nuestra parte del programa, que procesa las acciones de inspección del jugador, comprueba primero, si el objeto, sobre el cual se pide mas información, se encuentra cerca del jugador, o sea, en el mismo lugar, o si está incluso en su poder. En caso contrario, se puede proceder inmediatamente a la entrada de la próxima jugada:

```
5000 IF OB(N)<>JUGADOR AND OB(N)<>-1 TH
EN GOTO 5900
5900 REM OBJETO NO PRESENTE
5990 PRINT "AQUI NO VEO NADA PARECIDO !
": GOTO 1080
```

Además de estas dos condiciones, pueden formularse otros supuestos válidos como criterio para la ejecución de una acción, y sin los cuales una aventura no podrá llevarse a cabo:

Objeto está en el lugar
Objeto es llevado por el jugador
Objeto no está en el lugar
Flag activado
Flag desactivado
Jugador está en un lugar determinado

Como ya veremos más adelante, esta lista no exige integridad. Puede imaginarse por ejemplo, una inversión de la última condición,en la que determinadas acciones no puedan ejecutarse en un lugar concreto, pero si en cualquier otro lugar.

Los flags aparecidos en este momento requieren una explicación. Se trata de interruptores de señal, pero que en este caso sirven directamente al control de proceso del juego, y que queremos realizar con ayuda de un campo adicional. También estas variables de campo deben poder adoptar sólo dos estados o valores: o está activada la bandera o flag, en cuyo caso el contenido de la variable debe ser -1, o esta desactivada, lo que debe corresponder a un cero.

En la práctica, los usaremos para identificar determinados estados:

¿una puerta está abierta o cerrada?
¿un obstáculo ha sido eliminado o no?
¿un monstruo debe poner fin a las andanzas del jugador, o puede éste continuar el juego?

En la práctica de programación, sin embargo, la formulación de una única condición no suele bastar para clasificar la situación de manera inconfundible. Se requieren combinaciones lógicas de varias condiciones para permitir, que algunas acciones no sean ejecutadas incoherentemente y en un momento inadecuado.

Al hablar de estas operaciones lógicas, se trata de las combinaciones Y (AND) y O (OR). Referente a nuestra práctica de aventuras, esto significa, que deben cumplirse o ambas (AND), o sólo la una o la otra (OR) de dos o más combinaciones. Si no podemos renunciar a usar más de dos condiciones, tampoco podremos evitar los paréntesis. Un ejemplo debe ilustrarlo:

Pensemos en el explosivo en nuestar caja. Debe ser utilizado para quitarle al jugador la alegría de su rápido avance.

Sin embargo, para jugar limpio es conveniente dar un aviso antes de que se produzca la explosión, emitiendo tras la instrucción "Inspecciona explosivo" la información "Parece ser muy explosivo". Si el jugador no desiste en cogerlo, a pesar de este aviso, nadie más que el mismo tendrá la culpa de su triste final.

Si el jugador ha efectuado realmente esta entrada, el motor pone a disposición los números 1 (inspecciona) y 9 (explosivo). Ahora nuestro problema está en que no podemos darnos por satisfechos con la inspección del lugar. No debemos olvidar, que el jugador no puede coger el explosivo, pero si la caja de madera, y si lleva consigo la caja de madera, también tiene el explosivo. Comprobemos pues, si el jugador se ha referido al explosivo Y (AND) si, a la vez, se encuentra la caja de madera (objeto numero 7) en el lugar de la acción O (OR) en el inventario del aventurero.

El propio objeto de la acción (número 9) no está disponible, por lo que después de la línea 5000 se continua la ejecución del programa en la línea 5900.

Así que la ejecución de la instrucción ''Inspecciona explosivo'' debe formularse de la siguiente manera:

```
5904 IF N=9 AND (OB(7)=JU OR OB(7)=-1)
THEN PRINT "¡PARECE SER MUY EXPLOSIVO
!":GOTO 1080
```

LAS ACCIONES

El ejemplo anterior nos ha mostrado la ejecución de la acción más simple a realizar, que es la emisión de una información al jugador.

Pero esta instrucción PRINT sólo suele acompañar a otras acciones, como manipulaciones de OB(), para confirmar al aventurero la ejecución de su instruccion a través de un mensaje.

En contraposición a esto, instrucciones como Coge, Pon, Destruye, y también Abre significan siempre un cambio de posición para el objeto. Puede ser que éste sea llevado nuevamente con el jugador, o que ahora pertenezca al inventario de un determinado lugar; quizás incluso desaparezca completamente del juego. Imagínese el estupor del jugador de la fiebre del oro, si por lo visto el oso ya ha saboreado la miel, mientras que la botella todavía no ha aparecido en ningún sitio.

Otra acción también muy usada es el borrar o poner los ya mencionados flags (banderas), necesarios para crear progresivamente los supuestos de posteriores acciones.

Tampoco podemos olvidar en ningún caso, los condicionados cambios de paradero del jugador, que son causados por arte de magia o por acciones inofensivas. Una posterior ampliación de la fiebre del oro podría convertir la cueva del oso en un inmenso laberinto de pasillos y que, después de entrar "Inspecciona cueva", el jugador se encontrase en medio de la cueva, aunque no hubiese entrado en ella moviendose en una determinada dirección. Pero, incrementando el trabajo de programación, nuestra aventura a sólo resultara ser más realista, porque ¿ cómo puede inspeccionarse detenidamente una cueva, sin entrar antes en ella ?

De manera parecida que con las condiciones, también aquí podemos confeccionar una lista de posibles acciones, que tampoco en este caso pretende ser completa. Sin embargo, se trata de una base suficiente, ya que es posible escribir buenas aventuras utilizando solo las condiciones y acciones presentadas.

Emitir información al jugador
Objeto desaparece
Objeto entra en inventario
Objeto aparece nuevamente en el lugar
Flag es activado
Flag es desactivado
Jugador es trasladado a otro lugar

PROGRAMACION DE LA EJECUCION DE INSTRUCCIONES

A continuación encontrará indicaciones para ciertas particularidades, que deberán tenerse en cuenta al efectuar cada instrucción. Sólo le explicaré algunas líneas típicas del programa, usted tendrá que extraer las restantes líneas del listado de la miniversión de la fiebre del oro al final de este capítulo.

Acción: Inspecciona

De momento está claro, que el resultado de la inspección de un objeto siempre se comunica en forma de frase.

A menudo se tratará de una frase que siempre se repite (No veo nada especial). Por esta razón, no emitimos directamente la información para el jugador, sino que asignamos el texto a una variable al iniciar el programa y emitimos el contenido de ésta en el momento preciso.

Puesto que nuestro editor sólo admite sin problemas 120 caracteres para efectuar la entrada de líneas Basic, y que, por otra parte, además de las informaciones, estas líneas también deben contener las condiciones, en algunos casos no podremos evitar tener que preparar también algunos mensajes que sólo serán requeridos una vez.

```
600 REM ************************MENSAJES
601 M1$="NO VEO NADA ESPECIAL."
602 M2$="NO TENGO TANTA FUERZA."
603 M3$=" COMO TE IMAGINAS ESTO?"
604 M4$="EL OSO COGE LA MIEL Y DESAPARE
CE EN"
605 M5$="EL FONDO DE LA CUEVA."
```


Ya hemos explicado anteriormente la línea 5000, la cual asegura, que mediante la comprobación del valor del objeto actual en OB(), el mencionado objeto se encuentra cerca del jugador.

Si esto se cumple, se averiguará la línea de programa que se ocupa de este objeto según el número de objeto N. Si allí se cumplen todas las condiciones programadas, se procesará el resto de la línea y después se continuará la ejecución del programa con la nueva estructuración de la pantalla:

```
5002 IF N=1 THEN PRINT M1$:GOTO 1080
5003 IF N=3 THEN PRINT M1$:GOTO 1080
5004 IF N=4 THEN PRINT"EN UN RINCON HAY
UNA ESTANTERIA.":OB(8)=JUGADOR:GOTO 10
80
```

Si se inspecciona la cabaña de madera (N=4), el jugador descubre la estantería. Hasta ahora, ésta estuvo en el lugar 0, el almacén, y ahora es posicionada en el lugar donde se encuentra el jugador. Por ello, el motor de la aventura emitirá, entre otras cosas, ''una estantería desvencijada'' al procesar las líneas 1110 a 1200.

Pero también puede ocurrir, como en el caso de la estantería, que aparezca un objeto que el jugador pueda llevarse consigo.

Entonces ya no bastaría una formulación como la efectuada en la línea 5004, puesen cada posterior inspección de la estantería, la botella sería extraida automáticamente del inventario y colocada otra vez en el lugar.

Sólo podemos evitar este error, asegurándonos previamente, de que la botella aún no había aparecido en el juego, esto significa que todavía se encuentra en el almacén.

```
5008 IF N=8 AND OB(5)=0 THEN PRINT"EN L
A ESTANTERIA HAY UNA BOTELLA FORRADA."
```

```
:OB(5)=JUGADOR:GOTO 1080
5009 IF N=8 AND OB(5)<>0 THEN PRINT M1$
:GOTO 1080
```

Aún en inspecciones posteriores, el jugador ya no descubrirá nada especial (línea 5009).

Otro ejemplo debe mostrar el uso de los flags. Cuando el jugador haya encontrado el arca del tesoro, ve que está se halla cerrada con una cadena de hierro. Después de eliminar este obstáculo, no podrá verse su valioso contenido, sin antes abrir la arca. Para aclarar estos tres estados al jugador, sería posible introducir más objetos, tales como un arca cerrada con una cadena de hierro, un arca no cerrada y un arca abierta. Sin embargo, esta programación bastante laboriosa, que por otra parte también limita la memoria disponible, puede evitarse mediante el uso de interruptores. Lo más práctico es confeccionar unas tablas durante la construcción del juego de aventuras, para que cambie realmente el flag correcto:

Flag	0	-1
1	cadena entera	cadena destruida
2	arca cerrada	arca abierta

Ahora usted mismo podrá entender, por qué se ha previsto solo 0 y -1 como contenido, y por que no un único flag con diferentes cifras para cada estado.

Naturalmente nos servimos del hecho de que una expresión verdadera siempre se representa mediante un -1 para el interpretador Basic. Este hecho nos permite acortar las líneas de programa, en las instrucciones IF no necesitaremos indicar valores comparativos:

```
IF FL(1)=-1  corresponde a  IF FL(1)
IF FL(1)= 0  corresponde a  IF NOT FL(1)
```


Para nuestro ejemplo valen, pues, las siguientes líneas:

```
5011  IF N=11 AND NOT FL(1) THEN PRINT "
ESTA CERRADA CON UNA CADENA DE HIERRO."
:GOTO 1080
5012 IF N=11 AND FL(1) AND NOT FL(2) THE
N PRINT "POR FUERA NO VEO NADA ESPECIAL
.":GOTO 1080
5013 IF N=11 AND FL(1) AND FL(2) AND OB(
12)=0 THEN PRINT "ESTA LLENA DE MONEDAS
DE PLATA.":OB(12)=JUGADOR:GOTO 1080
```

Otro punto, al que deberemos dedicar especial atención, tiene su base en la interpretación de nuestro análisis de palabras.

Los objetos 1 y 2, completamente idénticos, pertenecen a la decoración de los dos primeros lugares. Si se comprueba la entrada efectuada por el jugador, sólo podrá determinarse un 1 como número de objeto, lo cual no responde a los hechos cuando el jugador se encuentre en el lugar 2.

Sin embargo, habíamos introducido la línea 5000 para poder detectar correctamente los errores de entrada, que ahora también nos facilita el tratamiento de estos casos excepcionales.

Así que solucionamos de la siguiente forma el problema ''Inspecciona árbol'' para el lugar 2:

```
5901 IF N=1 AND JUGADOR=2 THEN PRINT M1$
:GOTO 1080
```

Para finalizar esta rutina, debemos prever una última línea que será ejecutada sin ninguna condición, si no se han cumplido todas las condiciones necesarias en alguna de las

precedentes líneas de programa. Con ello se interceptan las entradas no definidas del juego, y evitamos que si no se ha encontrado ninguna condición correspondiente, se procesen las líneas de los restantes verbos, perdiendose entonces el control de la ejecución del programa. Incluso si olvidamos una acción innecesaria para la ejecución del juego de aventuras, el jugador nunca lo descubrirá:

```
5990 PRINT "¡AQUI NO VEO NADA PARECIDO !"
:GOTO 1080
```

Acción: Coge

Al igual que un jugador sólo puede inspeccionar un objeto que se encuentre cerca de él, también podrá cogerlo sólamente si se encuentra en el mismo lugar que el jugador, o si ya se encuentra en los bolsillos de éste.

```
6000 IF OB(N)<>JUGADOR AND OB(N)<>-1 TH
EN GOTO 6900
```

Según esto, sería muy fácil ejecutar esta orden. Sin embargo, deberíamos negarle esta posibilidad al aventurero para algunos objetos.

Así pues, tendremos en cuenta que un arca de hierro repleta es probablemente demasiado pesada para una sola persona. Tampoco podremos no hacer caso aún a intentos del jugador tan absurdos como ''Coge árbol''.

```
6001 IF N=1 THEN PRINT M2$:GOTO 1080
6005 IF N=11 THEN PRINT M2$:GOTO 1080
```

Naturalmente aún puede suceder algo peor, ya que debemos ponerle al jugador una cantidad suficiente de trampas como para no facilitarle una rápida victoria. Según nuestro guión,

el contacto con el explosivo debía producir una explosión y provocar el fin juego. Lo mismo sirve en el caso de que intente apoderarse de las pepitas de oro sin antes haber ofrecido un bocado exquisito al oso. Igualmente, el intento de coger el oso también debería resultar fatal para el jugador:

```
6015 IF N=1 AND JUGADOR=5 THEN MO$="EL
OSO ME HA MATADO .":GOTO 4500
6018 IF N=17 AND NOT FL(3) THEN MO$="UN
OSO SE LANZA SOBRE MI.": GOTO 4500
6900 IF N=9 AND (OB(7)=JUGADRO OR OB(7)
=-1) THEN MO$=""EL EXPLOSIVO HA ESTALL
ADO CON EL CON-":PRINT"TACTO !":GOTO 45
00
```

Nota: En la línea 4500 empieza una rutina que será llamada en caso de no ganar el juego. En esta rutina, MO$ esta prevista para una información aclaratoria.

El flag 3 es activado en el momento, en que el oso se apodere de la miel.

Si no hay nada que evite que el jugador se apodere de cualquier objeto de nuestra aventura, sólo debemos asignar al objeto correspondiente el nuevo número de situación, un -1:

```
6010 IF N=5 THEN OB(5)=-1:PRINT "O.K.":
GOTO 1080
```

Acción: Pon

En algún momento, el jugador querrá volver a deshacerse de estos objetos, bien porque quiera utilizarlos bien porque nosotros, como programadores, le hayamos limitado su capacidad de carga.

En la práctica, la rutina Pon no presenta apenas problemas. Además, es convincente por ser tan corta.

La única condición, es, que el objeto en cuestión debe estar en posesión del jugador, supuesto que se comprobará en la línea 7000:

```
7000 IF OB(N)<>-1 THEN PRINT "¿ PERO SI
NO TENGO NADA DE ESTO !":GOTO 1080
7900 OB(N)=JUGADOR:PRINT"O.K.":GOTO1080
```

Los errores producidos por el jugador son detectados inmediatamente y registrados correspondientemente. Si el objeto se encontraba en el inventario, se le asignará un nuevo paradero, modificando el contenido de OB().

En principio hay suficiente con estas dos líneas. Sin embargo, puede ser necesario efectuar otras acciones además del cambio de posición. En fiebre del oro, por ejemplo, habíamos previsto, que el oso se llevase la botella y desapareciese con ella en las profundidades de la cueva.

Naturalmente, esta acción sólo puede ejecutarse, si el jugador se encuentra en el lugar 5 en este momento. En cualquier otro lugar, esta orden también será ejecutada con la línea 7900.

```
7020 IF N=5 AND JUGADOR=5 THEN OB(5)=0:
FL(3)=-1:PRINT M4$:PRINT M5$:OB(14)=0:
GOTO 1080
```


Acción: Abre

Ya al principio de esta parte del programa se encuentra el conocido test que sirve para evitar errores técnicos. Por otra parte es necesario comprobar tanto el lugar como el inventario. Al fin y al cabo, no es necesario que el jugador coja una puerta antes de poder abrirla, condición que puede ser imprescindible cuando se trate de una botella:

```
8000 IF OB(N)<>JUGADOR AND OB(N)<>-1 TH
EN PRINT "AQUI NO HAY NADA PARECIDO.":
GOTO 1080
```

La propia acción suele constar de la activación de un flag y de la emisión de un mensaje:

```
8025 IF N=11 AND FL(1) THEN PRINT "O.K.
- LA TAPA CAE HACIA ATRAS.":FL(2)=-1:G
OTO 1080
```

Tampoco podemos olvidar acciones sin importancia, pero que son posibles tal como están las cosas. Como ejemplo puede servir otra vez nuestra botella. Un aventurero que piense con lógica puede querer abrirla antes de inspeccionarla más de cerca, y seguramente se sorprenderáa mucho, si no puede conseguirlo sólo porque nosotros lo consideramos improcedente en relación a la posterior marcha del juego:

```
8010 IF N=5 THEN PRINT "O.K":GOTO 1080
```

El final de este bloque permite que, además de entradas técnicas erróneas, también puedan detectarse imposibles en cuanto al contenido, tales como "Abre miel".

```
8999 PRINT "NO ENTIENDO LO QUE QUIERES
DECIR.":GOTO 1080
```

RESUMEN

Espero que estas explicaciones sobre las acciones más corrientes le hayan servido para entender la ejecución de la propia programación de juegos. En principio, se puede resumir afirmando, que cada acción aislada está encerrada entre dos líneas de programa.

Así se comprueba, al principio de un bloque, si la ejecución de la acción es posible desde el punto de vista técnico. Para finalizar, se asegura que la marcha del programa pueda continuar correctamente aún cuando aparezca cualquier error.

El reconocimiento de las correctas condiciones para la ejecución de una orden suele ser, al igual que la propia acción, una cuestión fácilmente programable, que además ha sido estandarizada en nuestro sistema de aventuras. El siguiente esquema deberá servirle de ayuda para realizar su primer juego de aventuras:

CONDICION	EN PROGRAMA BASIC
Objeto está en el lugar	OB(objeto)=JU
Objeto es llevado por jugador	OB(objeto)=-1
Objeto no está en el lugar	OB(objeto)<>JU
Flag está activado	FL(x)=-1
Flag no está activado	FL(x)=0
Jugador está en un lugar determinado	JU=número lugar

ACCIONES	EN PROGRAMA BASIC
Emitir información para el jugador	PRINT" o M$
Objeto desaparece	OB(objeto)=0
Objeto entra en inventario	OB(objeto)=-1
Objeto aparece de nuevo en el lugar	OB(objeto)=JU
Se activa flag	FL(x)=-1
Se desactiva flag	FL(x)=0
Jugador es trasladado a otro lugar	JU=número lugar

LOS ULTIMOS PASOS

Antes de que nuestro juego de aventuras responda a las imaginaciones desarrolladas en el capítulo 2, aún debemos hablar de otras tres partes del programa.

Para una marcha correcta del juego, es imprescindible proporcionar al jugador la posibilidad de efectuar un rápido inventario.

Será absolutamente posible efectuar esta acción de la misma manera que el tratamiento de los verbos restantes. Sin embargo, ya que en este caso se trata de una función básica de los programas de aventuras, ampliaremos nuestro motor de forma adecuada.

Inventario

El inventario pertenece a las llamadas instrucciones de una palabra, al igual que las rutinas de grabación y de carga del estado del juego, que se confeccionarán más adelante. Hemos dejado bastante espacio para esta parte del programa entre las líneas 1480 (final de la rutina de movimiento) y 2000 (análisis de la entrada).

Para evitar que nuestro motor este ocupado un tiempo innecesario en comparar cada jugada con las órdenes de este grupo especial, decidimos primeramente, si se trata de una de estas instrucciones especiales o de una entrada regular.

La entrada efectuada normalmente por el jugador tiene una longitud mayor de ocho letras. Por ello partimos de la suposición, de que se trata de una orden de una palabra si la longitud de la entrada es inferior:

```
1480 IF LEN(ENTRADA$)>8 THEN GOTO 2000
```

Si la entrada del jugador no está formada por una combinación regular de verbo/objeto, entonces se repasarán secuencialmente todas las líneas donde comience el tratamiento de una instrucción de una palabra, hasta que finalmente se determine la coincidencia de las tres primeras letras.

En la práctica, la ejecución del inventario se compone de un simple bucle, que muestra en la pantalla todos aquellos objetos del juego de aventuras cuyo paradero está señalizado con -1:

```
1499 REM **************INICIO INVENTARIO
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
2000
1510 PRINT"LLEVO CONMIGO LO SIGUIENTE:"
1520 FOR I=1 TO AO
1530 IF OB(I)=-1 THEN GOSUB 300+I:PRINT
T$
1540 NEXT I
1550 GOTO 1080
1560 REM ****************FIN INVENTARIO
```

Los títulos

Desde el punto de vista puramente técnico, con estas líneas ha terminado la confección de nuestro programa. Ya disponemos de todas las funciones deseadas y podemos empezar a jugar una partida. Pero, ¿cuánto tiempo debe vagar el aventurero por nuestro mundo? ¿Cuándo llegará a la meta?

En cuanto a las acciones, la miniversión de fiebre del oro acabará, cuando tanto las pepitas de oro como las monedas de plata están en posesión del jugador.

En este caso, basta una única línea Basic para impedir cualquier entrada posterior y dirigirse a una parte del programa, donde se informa al jugador de su victoria y da por finalizado el juego.

```
1340 IF OB(12)=-1 AND OB(17)=-1 THEN GO
TO 4800
```

Sin embargo, los juegos de aventuras suelen tener un final muy distinto, menos glorioso. Puesto que también muchos de los jugadores de nuestros programas se verán obligados a finalizar la partida antes de lo previsto, programaremos una nueva imagen del final a partir de la línea 4500.

Al ejecutar las acciones que finalizan el juego, ya habíamos preparado un mensaje adecuado, de manera que el jugador no se quedase sin conocer la causa de su fracaso.

```
4500 GRAPHICS 0:REM **** JUGADOR MUERTO
4600 PRINT ''¡ SOLO FALTABA ESTO !'':PRIN
T :PRINT MO$
4610 PRINT :PRINT ''¡ ESTOY MUERTO !'':PR
INT
4620 PRINT "  QUIERE QUE LO INTENTE OTR
A VEZ ";:INPUT ENTRADA$
4630 IF ENTRADA$(1,1)="S" THEN CLR :GOT
O 100
4640 GOTO 1960
```

La inclusión de la información individual en MO$ nos permite utilizar esta imagen titular en cada situación y en cada aventura. Sería muy poco laborioso efectuar modificaciones, en caso de que éstas fueran necesarias. Por esta razón, las líneas mencionadas, al igual que el mensaje victorioso a partir de 4800, pueden ser consideradas como ampliación adicional de nuestra aventura.

Además tengamos en cuenta, que no es normal cargar y ejecutar juegos de aventuras en poco tiempos.

Programemos , para terminar, final de programa regular.

Aunque este software se encuentre en el mercado, el hecho de poder salir del programa pulsando simplemente la tecla Break o Reset, no demuestra precisamente el fácil manejo del mismo.

Por ello introducimos otra instrucción de una sola palabra en nuestro sistema:

```
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
1950
1950 IF ENTRADA$(1,3)<>"FIN" THEN GOTO
2000
1960 GRAPHICS 0:PRINT ''¡ EL AUTOR LE DE
SEA MAS SUERTE PARA'':PRINT ''¡LA PROXIMA
VEZ!'':PRINT :PRINT :PRINT :END
```

En cuanto a su significado, éstas son las últimas líneas de nuestro programa. Hemos realizado todas las versiones imaginables sobre el final del juego. Pero, ¿cómo será el principio de nuestra obra?

Si ejecutamos nuestro programa tal como está en este momento, su primera acción consistirá en preparar las variables. Después nos encontraremos en pleno programa y es posible, que un inexperto aficionado a las aventuras entienda que el sentido del juego consiste en realizar una excursión ("sightseeing tour") a través de un mundo simulado electrónicamente.

Por ello tengamos en cuenta, que la primera impresión suele ser decisiva.

Pues, antes de dar por terminado nuestro programa, nos tomaremos la molestia de equiparlo con uno o varios títulos.

El primer título tendrá la misión de explicar escueta pero inteligiblemente, sobre los bytes que se encuentran en la memoria, nombrar título y clase del programa, y nombrar a los autores del mismo.

La próxima imagen será emitida en pantalla antes de inicializar las variables. Durante este período de tiempo, este título puede dar información sobre el objetivo que deberá alcanzar el jugador, e introducirlo en la acción del juego.

Sin embargo, si la solución del objetivo a alcanzar pertenece a la idea del juego, faltará este letrero, y seguidamente se añadirá al título una explicación de los juegos de aventuras, que en otro caso deberían implementarse en forma de una tercera imagen titular.

¿Se acuerda cómo fue, cuando usted tuvo la primera aventura en sus manos y en la memoria del ordenador? ¿Conoció el sentido y objetivo de estos programas, o estuvo sentado delante de su ordenador sin saber que hacer?

No olvidemos, que siempre habrá principiantes. Principiantes que se alegran, si después de enchufar el ordenador pueden cargar e iniciar un programa sin que aparezcan mensajes de error. Al fin y al cabo, programar una instrucción básica de juego sólo requiere un poco de trabajo adicional. Vayamos pues a efectuarlo, y no excusemos la falta de estas instrucciones diciendo que el descubrimiento de las mismas ya supone el primer obstáculo del juego.

A tal fin, disponemos de las líneas de programa 10 - 100, así como de 700 - 900

Para que sea más razonable, introducimos las explicaciones generales en un subprograma, que nos permite la posterior ampliación del motor de la aventura con el comando INStrucciones. El aventurero puede reclamar ahora las reglas generales de juego siempre que quiera, sin necesidad de interrumpir la partida.

ADVERTENCIAS SOBRE EL SIGUIENTE LISTADO

Después de haber entrado las siguentes líneas en su Atari, usted dispondrá de la miniversión de fiebre del oro capaz de funcionar. Pero debe tener en cuenta, que sólo se trata de líneas adicionales de programa, y que todas las líneas explicadas en el texto anterior deben hallarse ya en la memoria.

Usted verá que, con ayuda de las técnicas que le hemos presentado hasta ahora, ya podrá desarrollar sofisticados juegos de aventuras, que no tienen nada que envidiar al correspondiente software que puede adquirirse en los comercios.

Depende de usted, si por ahora desea dejar de lado este libro y desarrollar un propio juego de aventuras, o si prefiere conocer más detalles qué puedan transformar una buena aventura a una de perfecta.


```
1 REM FIEBRE DEL ORO,MINIVERSION;ATARI
2 REM (C) WALKOWIAK, OCTUBRE 1984
3 REM ----------------------------
10 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
4,1,0:FOR I=1 TO 9:PRINT :NEXT I
20 PRINT "
  "
30 PRINT "    F I E B R E   D E L   O
R O"
40 PRINT "
  "
50 PRINT "            MINIVERSION 1.0
  "
60 PRINT "         (C) 1984 BY WALKOWIA
K "
70 PRINT "
 ";
80 FOR I=1 TO 3000
90 NEXT I
100 REM ******** DATOS DE LA AVENTURA
110 AR=6:AO=18
150 DIM LUGAR$(80),PASADA(6,6),DIRECCI
ON$(36),ENTRADA$(20),T$(40),OB(AO),BLA
NCOS$(38),DZ$(50),ZENDE$(3),FL(5)
160 DIM VERBOS$(40),EVERBO$(20),EOBJET
O$(20),OBJETOS$(120),MO$(40),M1$(30),M
2$(30),M3$(30),M4$(38),M5$(38)
190 GOTO 600
318 T$="UNA BARRA DE HIERRO":RETURN
410 DATA 1,2,2,3,0,0,3,0,0,0,0,0,5,0,4
,6,5,0
699 REM ******* TITULO 2: INTRODUCCION
700 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
 4,1,0
710 PRINT "Bienvenido a la miniversion
 de":PRINT "FIEBRE DEL ORO!":PRINT
720 FOR I=1 TO 37:PRINT CHR$(13);:NEXT
 I:PRINT
730 PRINT "Hace unos dias, y mientras
probaba":PRINT "suerte en un mundo nue
```

```
vo, Ud encontro"
740 PRINT "a un viejo moribundo al que
 atendio":PRINT "en las ultimas horas
de su vida."
750 PRINT "Agradecido por el acto, el
viejo le"
760 PRINT "hablo de una mina de oro qu
e poseia y":PRINT "de parte de la fort
una que alli tenia"
770 PRINT "escondida.":PRINT "Ud ha su
perado infinidad de peligros"
780 PRINT "hasta dar con ella; pronto
alcanzara":PRINT "su objetivo y podra
comprobar, si lo"
785 PRINT "que el viejo decia era cier
to, o si":PRINT "solo se trataba de un
 sueno delirante":PRINT
790 FOR I=1 TO 37:PRINT CHR$(13);:NEXT
 I:PRINT
795 PRINT "DESEA RECIBIR CONSEJOS PARA
 CONTINUAR";:INPUT ENTRADA$
798 IF ENTRADA$(1,1)="S" THEN GOSUB 80
0
799 GOTO 900
800 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
 4,1,0
820 PRINT "              ATARI - VENTURES
          ":PRINT "      (C) 1984 BY JO
ERG WALKOWIAK      ":REM INVERS
830 PRINT :PRINT "Imaginese un robot a
l que puede con-  trolar con numerosos
 comandos. Yo soy"
840 PRINT "este robot y me pondre en s
u lugar"
850 PRINT "para enfrentarme a todos lo
s peligros de las mas excitantes avent
uras."
855 PRINT "Para que pueda manejarme co
modamente, le describire en cada momen
to la si-"
```



```
860 PRINT "tuacion en que me encuentre
.":PRINT "A continuacion, Ud me indica
 con dos"
863 PRINT "palabras, como por ejemplo
INSPECCIO- NA PUERTA, COGE CUCHILLO, l
o que deba":PRINT "hacer."
870 PRINT :PRINT "Ademas entiendo las
ordenes:"
880 PRINT "INVENTARIO y FIN.":PRINT
890 PRINT "POR FAVOR, PULSE <RETURN> Y
 ...";:INPUT ENTRADA$:GRAPHICS 0:RETUR
N :REM INVERS
1000 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLO
R 4,1,0:GOTO 1020
1030 T$=""
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
 1900
1900 IF ENTRADA$(1,3)<>"INS" THEN GOTO
 1950
1910 GOSUB 800
1920 GOTO 1080
1999 REM ********** SEPARAR ENTRADA
2060 VN=0:N=0
2070 EVERBO$=EVERBO$(1,3)
2080 EOBJETO$=EOBJETO$(1,3)
4800 GRAPHICS 0
4810 PRINT "ENHORABUENA!"
4820 PRINT :PRINT :PRINT "HA RESUELTO
EL PROBLEMA PLANTEADO":PRINT :PRINT "Y
 PUEDE AFRONTAR UNA NUEVA AVENTURA."
4840 PRINT :PRINT :PRINT :END
4999 REM * EJECUTAR JUGADAS DE JUGADOR
5005 IF N=5 THEN PRINT "LA BOTELLA EST
A LLENA DE MIEL.":GOTO 1080
5006 IF N=6 THEN PRINT M1$:GOTO 1080
5007 IF N=7 AND OB(9)=0 THEN PRINT "EN
 LA CAJA HAY EXPLOSIVO.":GOTO 1080
5010 IF N=10 THEN PRINT "EN EL AGUJERO
 HAY UN ARCA DE HIERRO.":OB(11)=JUGADO
R:GOTO 1080
```

```
5014 IF N=12 THEN PRINT "ES JUSTO LO
UE BUSCABA!!":GOTO 1080
5015 IF N=13 THEN PRINT "HE ASUSTADO
 UN OSO.":OB(14)=JUGADOR:GOTO 1080
5017 IF N=15 THEN PRINT "EN MEDIO HAY
UN AGUJERO.":OB(10)=JUGADOR:GOTO 1080
5018 IF N=16 THEN PRINT "PARECEN SER
UY RESISTENTES.":GOTO 1080
5019 IF N=17 THEN PRINT "SE TRATA DE
RO PURO!":GOTO 1080
5902 IF N=6 AND OB(5)=-1 THEN PRINT "
S DULCE Y SABROSA.":GOTO 1080
5903 IF N=6 AND OB(5)<>-1 THEN PRINT
NO TENGO MIEL!":GOTO 1080
5910 IF N=1 AND JUGADOR=5 AND OB(14)=
 THEN PRINT "PARECE QUE DESPIERTO SU
PETITO.":GOTO 1080
6002 IF N=3 THEN PRINT M2$:GOTO 1080
6003 IF N=4 THEN PRINT M3$:GOTO 1080
6004 IF N=8 THEN PRINT M2$:GOTO 1080
6006 IF N=10 THEN PRINT M3$:GOTO 1080
6007 IF N=13 THEN PRINT M3$:GOTO 1080
6008 IF N=15 THEN PRINT M2$:GOTO 1080
6011 IF N=6 THEN OB(5)=-1:PRINT "O.K.
:GOTO 1080
6012 IF N=7 THEN OB(7)=-1:PRINT "O.K.
:GOTO 1080
6014 IF N=12 THEN OB(12)=-1:PRINT "O.
.":GOTO 1080
6016 IF N=16 THEN OB(18)=-1:PRINT "O.
.":GOTO 1080
6017 IF N=17 AND FL(3) THEN PRINT "O.
.":OB(17)=-1:GOTO 1080
6999 PRINT "AQUI NO VEO NADA PARECIDO
":GOTO 1080
7010 IF N=6 AND JUGADOR=5 THEN OB(6)=
:FL(3)=-1:PRINT M4$:PRINT M5$:OB(14)=
:GOTO 1080
8005 IF N=4 AND JUGADOR=3 THEN PRINT
LA CABANA YA ESTABA ABIERTA.":GOTO 10
```



```
0
8020 IF N=11 AND  NOT FL(1) THEN PRINT
 "LA CADENA NO LO PERMITE!":GOTO 1080
9000 REM ********** VERBO=UTILIZAR
9010 IF N=16 AND JUGADOR=4 THEN PRINT
"LA CADENA SE ROMPE.":FL(1)=-1:GOTO 10
80
9999 PRINT "NO ENTIENDO LO QUE QUIERES
 DECIR!":GOTO 1080
10000 IF N=18 AND JUGADOR=4 AND OB(18)
=-1 THEN PRINT "LA CADENA SE ROMPE.":F
L(1)=-1:GOTO 1080
10010 IF N=16 AND JUGADOR=4 AND OB(18)
<>-1 THEN PRINT "NO TENGO NINGUNA BARR
A DE HIERRO.":GOTO 1080
10999 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR!":GOTO 1080
```

CAPITULO 4
- PERFECCIONAMIENTO -

DE LA BUENA A LA PERFECTA AVENTURA

Si usted ha jugado con la miniversión de fiebre del oro, o si usted mismo ha confeccionado una aventura partiendo de la base de las técnicas descritas hasta ahora, coincidirá conmigo en que se ha alcanzado perfectamente el estándar de programas corrientes de la clase aventuras.

Aún así pueden encontrarse a veces juegos de aventuras que ofrecen extras adicionales. Para poder superar la competencia, se simplificó su manejo, o se busco la forma de añadir también en la realización del juego pequeño detalles cotidianos, que lograban complicar más aún la vida del jugador.

También existen algunos programas, que con ayuda del correspondiente dispendio en publicidad, llaman la atención sobre sus características especiales que en realidad no existen.

También se activa el oído del jugador con aislados beeps y pips, que señalan una orden entendida o no ejecutable por medio del apoyo acústico de objetos visibles hasta la imitación de la voz humana. Desgraciadamente, la utilidad de estos extras es nula para el juego, por eso vale la pena si es necesario sacrificar su dinero por el afán de conseguir efectos especiales que llenan la memoria, o si prefiere un mundo de aventuras ampliado por la memoria ahorrada.

FACILIDAD DE MANEJO

Esta siempre se exige en los programas comerciales,¿ por ¡ué no también para los juegos?

Realmente existen juegos de aventuras que ni siquiera nforman al jugador sobre la manera en que puede salir de un lugar peligroso. Sólo están esperando llevarlo a la

desesperación con la aparición del mensaje ''No puedes ir en esa dirección. - You can't go in that direction.'', que emiten después de que el jugador haya efectuado entradas de diversos puntos cardinales.

Bien, es lógico que el jugador se desespere. Sin embargo, parece, que estos programas han sido escritos para atraer al aficionaado hacia las aventuras con el anuncio ''¡Le garantizamos, que necesitará meses para poder conocer todos los lugares de esta aventura!'' e inducirle a comprarlo.

Afortunadamente, esta crítica no puede afectar a nuestro sistema de aventuras. Sin embargo, la falta de cualquier posibilidad de poder grabar el estado actual de la partida, condiciona una pérdida progresiva del interés por el juego.

Desgraciadamente, esto ocurre con más frecuencia en aventuras cuya confección nos ha comportado más trabajo, y de cuya complejidad y trampas puestas de forma inteligente estamos especialmente orgullosos.

Echémos pues una mano al jugador, que de todos modos necesitará tener más de siete vidas, para conocer todas las formas posibles de morir de la aventura, creando además todas las condiciones técnicas necesarias para poder continuar el juego habiendo adquirido otra experiencia a partir de este momento. Si después, el jugador no hace uso de esta ampliación razonable en el debido momento, ya sólo puede enojarse consigo mismo.

Además, con estas rutinas evitamos un corte en la propia carne, ya que durante la fase de verificación de un juego de aventuras se presentará repetidas veces la necesidad de introducir pequeños cambios de programa, que desgraciadamente también aniquilan los contenidos de todas las variables.

Entonces la única solución que nos queda es empezar de nuevo con Run y repetir todas las entradas necesarias para volver a entrar en el lugar donde se produjo el error, y para llegar a la conclusión de la incapacidad de esta nueva versión para superar a la anterior.

SAVE GAME / LOAD GAME

Pensemos primero en que se diferencia una determinada escena del juego de la posición inicial.

Inmediatamente recordamos los movimientos del jugador de manera más o menos definida dentro nuestro mundo. También pensamos en los objetos que éste ha recogido, colocado, destruído, gastado o comido, o sea, en objetos que ya no existen, así como en objetos que han aparecido recientemente en el juego.

Además de estos contenidos de variables visibles, y quizás menos importantes por ser controlables, también debemos salvar las modificaciones hechas para controlar la marcha del juego. En caso contrario, puede ocurrir que una acción no alcance éxito deseado a pesar de disponer de todos los objetos necesarios para ello.

Pensemos solamente en los flags o banderas, que no sólo pueden usarse siempre para la marcha de una acción coherente, sino incluso para controlar sucesos aparentemente incoherentes.

Lo mismo sirve también para nuestra tabla de direcciones. Incluso aquí es perfectamente normal efectuar modificaciones substanciales, como veremos más adelante.

Dicho brevemente: Los contenidos de las variables JUGADOR, OB(), FL() y PASADA(,) deben grabarse en forma de fichero secuencial en un diskette o cinta antes de interrumpir el juego.

INCISO: MEMORIZACION EXTERNA DE DATOS

Además de la memorización interna de datos, donde los datos son depositados en la memoria de trabajo propia del CPU que, por consiguiente, siempre dispone de ellos; para tratar con datos de cualquier tipo también se necesitan memorias externas que amplían la memoria del ordenador casi indefinidamente, y que sirven esencialmente para archivar y disponer grandes cantidades de datos. Como miembros de esta familia pueden nombrarse por ejemplo, las tarjetas y cintas perforadas, cintas magnéticas, unidades de disco o cilindro, así como nuestro floppy 1050 o el cassete Atari 1010.

Además de las diversas capacidades y tiempos de acceso, también pueden establecerse diferencias dentro la forma de almacenar los datos.

Los equipos de memoria, tales como una perforadora/lectora de tarjetas o como un cassete de datos, trabajan con los llamados ficheros secuenciales, y pueden comparase más bien con un antiguo rollo de papiro, mientras que una unidad de disco o diskette suele trabajar con ficheros relativos, que le asemejan más a un fichero con fichas corrientes.

Ambos procedimientos tienen sus ventajas e inconvenientes, que se aprecian especialmente en su correspondiente cantidad de memoria ocupada y en el tiempo empleado en encontar un determinado registro.

Así, cada dirección de un fichero de clientes requiere

una ficha propia, mientras que en el rollo, para ahorrar papel, se anotaron todas las direcciones de la forma más reducida posible. Pero cuando hay que buscar la dirección del cliente 234, se coge simplemente la ficha número 234, ya que naturalmente están todas numeradas, mientras que con el rollo no puede evitarse tener que leer, o al menos contar desde el principio todas las direcciones hasta llegar a la 234.

Los lectores que se interesen más por los ficheros tendrán que consultar literatura al respecto. Por lo demás, debe bastar la mención de que el almacenamiento secuencial es justo el procedimiento correcto para nuestro problema, porque queremos memorizar los contenidos de nuestras variables uno tras otro y volver a cargarlos más tarde. Esto significa, que no necesitamos métodos especiales de acceso.

SAVE GAME

La transmisión de datos se efectuará por razones prácticas dentro de los bucles correspondientes, para lo que se exige, ampliar los datos particulares del programa por la cantidad de flags utilizados, pues también a este bucle se le debe poder asignar un valor final.

Además debemos proceder a modificar la línea 1500 del programa, para introducir la consulta de las instrucciones especiales en la nueva rutina:

```
110 AR=6:A0=18:AF=3

1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
1600
```

Si con la última entrada del jugador no se trataba con la

instrucción Inventario, se efectuarán algunos saltos hasta determinar la identidad con una de las instrucciones de una sola palabra:

```
1599 REM ********************* SAVE GAME
1600 IF ENTRADA$(1,3)<>"SAV" THEN GOTO
1700
1699 REM ********************* LOAD GAME
1700 IF ENTRADA$(1,3)<>"LOA" THEN GOTO
1900
```

Antes de que el sistema operativo de nuestro ordenador pueda transmitir los datos, se deberá equipar previamente con algunas informaciones, tales como la dirección de transmisión, el camino de transporte y dirección; misión que ejecuta la instrucción OPEN.

```
1620 OPEN#1,8,0,"D:GAME"
```

Después de ejecutar esta línea, se dispone de un fichero de datos en el diskette (D:), en el que ahora puede escribirse secuencialmente (8).

En el caso de que usted utilice cassette, sustituya "D:" por "C:".

Después de estos preparativos, ya no hay nada que impida la transmisión de datos. La salida de datos hacia el diskette se efectúa de forma parecida a una salida hacia la pantalla. Sólo debe desviarse la instrucción PRINT indicando un número de fichero (1).

Antes de volver al programa principal, una instrucción CLOSE final nos protegerá de errores que puedan producirse posteriormente.

```
930 FOR I=1 TO AF:FL(I)=0:NEXT I
```



```
1625 PRINT #1,JUGADOR
1630 FOR I=1 TO AO
1631 PRINT #1,OB(I)
1632 NEXT I
1635 FOR LUGAR=1 TO AR
1636 FOR DIRECCION=1 TO 6
1637 PRINT #1,PASADA(LUGAR,DIRECCION)
1638 NEXT DIRECCION
1639 NEXT LUGAR
1645 FOR I=1 TO AF
1646 PRINT #1,FL(I)
1647 NEXT I
1650 CLOSE #1
1670 GOTO 1080
```

LOAD GAME

Para reconstruir una situación determinada es necesario asignar nuevamente estos datos a las correspondientes variables, misión de la que se encarga una casi idéntica parte del programa:

```
1720 OPEN 1,4,0,"D:GAME"
1725 INPUT #1,JUGADOR
1730 FOR I=1 TO AO
1731 INPUT#1,SITUACION:OB(I)=SITUACION
1732 NEXT I
1735 FOR LUGAR=1 TO AR
1736 FOR DIRECCION=1 TO 6
1737 INPUT #1,META:PASADA(LUGAR,DIRECCI
ON)=META
1738 NEXT DIRECCION
1739 NEXT LUGAR
1745 FOR I=1 TO AF
1746 INPUT #1,FLAG:FL(I)=FLAG
1747 NEXT I
1750 CLOSE #1
1770 GOTO 1080
```

Naturalmente, si utiliza cassette, debe volver a transformar "D:" en "C:".

Para hacer nuestro programa más lujoso, le añadiremos algunas líneas más. Así que no es en absoluto necesario limitarnos a un único fichero para grabar. Un segundo fichero sera imprescindible, cuando otros miembros de nuestra familia muestren su interés por las aventuras y deseen probar su habilidad en el mismo juego.

Demos a los jugadores la libertad de elegir, y permitamos que ellos mismos escojan un nombre de 8 letras como máximo. El juego de aventuras deberá entonces señalar este fichero como fichero de juegos con el complemento DAT (FIChero).

Resulta útil evitar la interrupción de la ejecución del programa en el caso que aparezca un error, cosa que puede suceder fácilmente si se ha introducido un diskette incorrecto, o ninguno, en la unidad de disco.

Los subprogramas completados de esta manera serán parecidos al siguiente listado.


```
1599 REM ********************* SAVE GAME
1600 IF ENTRADA$(1,3)<>"SAV" THEN GOTO
 1700
1605 PRINT "CON QUE NOMBRE ";:INPUT EN
TRADA$:IF LEN(ENTRADA$)>8 THEN PRINT "
UN POCO MAS CORTO!":GOTO 1605
1610 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)
=".DAT"
1615 T$(LEN(T$)+1)=ENTRADA$
1620 OPEN #1,8,0,T$
1625 PRINT #1,JUGADOR
1630 FOR I=1 TO A0
1631 PRINT #1,OB(I)
1632 NEXT I
1635 FOR LUGAR=1 TO AR
1636 FOR DIRECCION=1 TO 6
1637 PRINT #1,PASADA(LUGAR,DIRECCION)
1638 NEXT DIRECCION
1639 NEXT LUGAR
1645 FOR I=1 TO AF
1646 PRINT #1,FL(I)
1647 NEXT I
1650 CLOSE #1
1670 PRINT "O.K.":GOTO 1080
1699 REM ********************* LOAD GAME
1700 IF ENTRADA$(1,3)<>"LOA" THEN GOTO
 1800
1705 PRINT "QUE JUEGO ";:INPUT ENTRADA
$:IF LEN(ENTRADA$)>8 THEN PRINT "ESTE
NO PUEDE EXISTIR!":GOTO 1705
1710 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)
=".DAT"
1715 T$(LEN(T$)+1)=ENTRADA$
1720 OPEN #1,4,0,T$
1725 INPUT #1,JUGADOR
1730 FOR I=1 TO A0
1731 INPUT #1,SITUACION:OB(I)=SITUACIO
N
1732 NEXT I
1735 FOR LUGAR=1 TO AR
```

EL LEXICO

A menudo, la elección adecuada del léxico para un juego de aventuras se convierte en una tortura, ya que el jugador está obligado a utilizar las mismas expresiones que el programador, y no siempre es fácil.

Normalmente, las importaciones que fueron desarrolladas para el mercado americano más avanzado, no permiten ser resueltas con el inglés aprendido en la escuela, o al menos resulta muy difícil. Por esta razón, muchos jugadores suelen tener cerca del ordenador su diccionario castellano - inglés, pero también este sistema puede fracasar cuando el autor ha empleado el lenguaje vulgar o algún argot.

Si observamos algunos fracasos de estos juegos debidos al lenguaje original,nos alegraremos con juegos editados en castellano.

Pero inmediatamente nos damos cuenta de las limitaciones de nuestro propio idioma. La primera de las dificultades se presenta al intentar adaptar los originales ingleses utilizando la misma construcción de la frase, es decir usando tan sólo dos palabras para cualquiera de las situaciones que puedan presentarse. Puede que el castellano hablado por los turistas sea aceptable para tal fin; incluso las más rebuscadas construcciones sintácticas pueden ser graciosas en un principio. Pero, ¿ cómo pueden ocurrírsele a una persona de disposición y talento normales este tipo de entradas sin que nadie le ayude?

Aunque una frase como "Coge cuchillo" no nos parezca del todo mal, debemos tener en cuenta que el cuchillo también puede realizar otras tareas. Sin embargo y a pesar de ello, sigue pareciéndonos un objeto agradable ya que frases como "Lanza cuchillo", o "Afila cuchillo" no exigen demasiada fantasía para entenderlas con claridad.

Cuando el cuchillo haya cumplido su misión, habrá que desprenderse de el tarde o temprano, hecho que en algunos juegos se da con la pérdida voluntria del mismo. ''Pon cuchillo'' causa mejor impresión que ''Pierde cuchillo'', pero automáticamente nos preguntamos donde. Aunque esta entrada no corresponda precisamente a las habituales reglas del juego, deberíamos tenerla en cuenta por lo menos en nuestro sistema.

Para finalizar, volvamos a observar un ejemplo de la primera página de este libro. ¿ Cómo podemos trepar a la cima del árbol? ¿ Quizás con ''Trepa árbol''?

Desgraciadamente no, pero ¿ dónde nos lleva un ''Sube árbol'' que suena mejor?

Sólo nos informa de que el verbo "Subir" no ha sido entendido. ¿ Qué hacer? Rendirse o continuar cavilando?

No podrá evitarse tener que probar con verbos como trepar, subir y escalar y, si la acción es realmente necesaria, tarde o temprano conseguiremos el éxito.

Naturalmente, los problemas de esta tipo no resultan muy agradables y, puesto que este capítulo trata del perfeccionamiento de nuestro sistema de aventuras, buscaremos soluciones más adecuadas.

SINONIMOS

La primera propuesta para solucionar el problema del lenguaje sería introducir sinónimos en cantidad suficiente. Sin embargo, como muestra el ejemplo del árbol, el equipamiento consecuente con sinónimos puede aumentar

considerablemente la extensión del programa, comporta una gran cantidad de trabajo adicional y una limitación de la memoria disponible, cada vez más escasa.

Aun así, introduciremos aquí un tipo de solución que aplicaremos como de costumbre a nuestro pequeño ejemplo de la fiebre del oro, que ya casi convence por su brevedad y sencillez. Naturalmente no se trata de líneas que sean exigidas necesariamente por el programa. Sólo depende de usted, si prefiere un vasto léxico o si se conforma con la propuesta que le presentamos seguidamente para mejorar la facilidad de manejo del programa.

La primera ocasión de no encontrar inmediatamente la palabra correcta, se le presenta al jugador de fiebre del oro en el lugar 3 al intentar inspeccionar más de cerca la cabaña de madera.

De momento fracasará con "Inspecciona choza", y triunfará con ''Inspecciona cabaña''.

Para que se pueda entender la segunda palabra, deberemos ampliar antes que nada nuestro vocabulario:

```
110 AR=6:AO=19
180 OBJETOS$="ARBO... ...CHOZ"
```

Podemos ahorrarnos la descripción detallada. Asimismo, el cero se hace obligatorio como cc signa, ya que no deberá aparecer otra cabaña ni en el inventario ni en un lugar.

Sólo se necesita el nombre para que el motor de la aventura pueda averiguar el correspondiente número de objeto, lo que sucede en la línea 2160.

Normalmente, la ejecución del programa continua en la línea 2200 con un salto hacia la correspondiente ejecución de instrucción. Por supuesto no hay una sola condición que se cumpla, pues en ningún lugar se pide un valor de 19 para N. Para no tener que completar estas

líneas, insertaremos el siguiente artificio:

```
2160 IF EO$=RN$(N) THEN 2185
2184 REM ********** SINONIMOS
2185 IF N=19 THEN N=4
```

con lo que se realizarán las mismas acciones con la choza de madera que con la cabaña de madera.

Para conseguir que nuestro programa también pueda entender mayor cantidad de verbos, el trabajo será menor quea antes.

Una palabra que inicia la misma acción estará también disponible en el léxico de la aventura, y simplemente habrá que completar la tabla de saltos con el destino de salto que ya había sido utilizado una vez.

De esta manera ampliamos fiebre del oro con la función "Observa" idéntica a "Inspecciona":

```
170 VERBOS$="INSP... ...OBSE"
2200 ON VN GOTO 5000,6000,7000,8000,900
0,10000,5000
```

Como puede ver, nuestro sistema es especialmente fácil de ampliar con verbos sinónimos, por lo que no deberíamos prescindir de este bono.

En cambio, podemos tratar con negligencia los sinónimos de objetos, ya que tampoco les corresponde el mismo significado. Casi siempre, con excepción de las aventuras gráficas, las líneas de encabezamiento indican puntos de referencia.

Todavía podemos facilitar mas las cosas al jugador, de manera que únicamente tenga que concentrarse en solucionar los problemas sin perder el hilo buscando las palabras correctas.

¿Qué le parecería a usted, si un juego de aventuras mostrase su vocabulario completo a petición del jugador? ¿ Y si confeccionase una lista de todos los verbos posibles?

VOCABULARIO

Un listado del vocabulario completo debería acabar con todos los problemas en la elección de palabras, y el jugador no perdería ni tiempo ni interés efectuando entradas inútiles.

Desgraciadamente, también aquí encontramos ventajas e inconvenientes.

Imaginemos un jugador que aún no conoce fiebre del oro. Aunque no haya jugado demasiado, podrá desarrollar en poco tiempo una estrategia de juego que lo lleve rápidamente al final, siempre y cuando sea medianamente capaz de pensar consecuente y de forma lógica. Desde el comienzo del juego conoce el completo vocabulario de la aventura y sabe, que en algún lugar podrá encontrar una botella, que en otro habrá una arca. Naturalmente también desarrollara una atracción hacia la plata y las pepitas de oro. Seguidamente pensará en los objetos que se pueden destruir, y su elección recaerá en la botella, la cabaña y la cadena.
Se esforzará en encontrar el arca del tesoro y, después de darse cuenta de que una cadena de hierro le impide ver el contenido, intentará destruirla, lo que difícilmente podrá conseguir sólo con ayuda de las manos. Otro VOK le dará la idea de que la barra de hierro es probablemente el único objeto capaz de cumplir realizar esta tarea.

Pero también los peligros serán de menor envergadura, si ya conoce, que comparte su mundo con un oso.

Así pues, también depende del programador valorar el hecho

de dar a conocer el vocabulario completo de su aventura o sólo una pequeña parte del mismo.

Esto podría realizarse con ayuda de una segunda imagen titular, que emitiera cierta cantidad de palabras importantes. Sin embargo, con ello actuaríamos en contra de nuestro objetivo de desarrollar un universal sistema de aventuras, ya que esta tabla tendrá que confeccionarse de nuevo para cada programa.

Sin embargo, en los grandes juegos de aventuras, con su correspondiente léxico, es perfectamente razonable efectuar un listado del vocabulario, por lo que tampoco queremos prescindir de ello al desarrollar nuestro sistema de aventuras.

Por las razones anteriormente mencionadas, el motor de la aventura se encarga del tratamiento de la instrucción VOC. Para permitir una aplicación realmente universal de esta parte del programa, hay que averiguar las palabras destinadas a ser emitidas de los datos de juego de cada aventura.

Pero de momento, nuestro sistema sólo ofrece las abreviaciones necesarias para poder identificar cada término, razón por la cual cambiaremos antes que nada los strings afectados. Para que pueda seguir funcionando la rutina que determina cada palabra aislada de estas variables de cadena VERBOS$ y OBJETOS$, la longitud de cada término debe ser la misma. Por ello, la longitud de palabra se regirá ahora por el número de letras de la palabra más larga:

```
170 VERBOS$="INSPECCIONACOGE      PON
ABRE      UTILIZA    DESTRUYE
OBSERVA    "
```

También se debe cambiar la longitud de paso de los bucles, que en lugar de cuatro será once. Pero si queremos evitar los cambios cada vez que realizamos un juego, introduciremos la nueva variable WL para significar la longitud de palabra:

```
130 WL=11

2090 FOR I=1 TO LEN(VERBOS$) STEP WL
2140 FOR I=1 TO LEN(OBJETOS$) STEP WL
```

A continuación de las rutinas INV, SAVE y LOAD añadiremos el correspondiente subprograma teniendo en cuenta la frecuencia de llamada de las líneas correspondientes:

```
1800 IF ENTRADA$(1,3)<>"VOC" THEN GOTO
1900
1805 GOSUB 1880
```

En principio, en la línea 1805 se llama a un subprograma que borra la pantalla y emite una imagen titular. La utilización de una rutina propia se justifica por la repetida necesidad de esta función en juegos de aventuras más extensos, cuando hay que presentar varias páginas una tras otra.

```
1880 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
4,1,0:PRINT "ENTIENDO LOS VERBOS SIGU
IENTES:":PRINT :RETURN
```

Después de determinar y emitir las palabras en cuestión, la línea 1890 espera que se pulse la tecla RETURN para reanudar seguidamente la ejecución del programa:

```
1815 FOR I=1 TO LEN(VERBOS$) STEP 11
1820 PRINT VERBOS$(I,I+10)
1830 NEXT I
```



```
1870 GOSUB 1890:GRAPHICS 0:SETCOLOR 2,1
,0:SETCOLOR 4,1,0:GOTO 1080
1890 PRINT :PRINT :PRINT "PULSAR <RETUR
N> ... ";:INPUT ENTRADA$:RETURN
```

Las líneas precedentes trabajarán ahora de la forma prevista, al menos mientras no haya que emitir más de dieciocho palabras. Al superar esta cantidad, el tiempo que emplea la emisión de los primeros vocablos en la pantalla será difícilmente suficiente como para considerarlo una ayuda al jugador.

Para solucionar este problema, utilizamos una variable de control, cuyo contenido se fija en cero al comienzo de la emisión, y que aumenta su valor en uno con la emisión de cada palabra. Si después de dieciocho palabras se ha escrito la correspondiente cantidad de líneas, entonces borramos la pantalla y volvemos a fijar el contador a uno.

```
1810 IMPRIMIDO=0
1820 PRINT VERBOS$(I,I+10):IMPRIMIDO =I
MPRIMIDO+1
11825 IF IMPRIMIDO=18 THEN GOSUB 1890:I
MPRIMIDO=1:GOSUB 1880$
```

De la misma manera, el jugador también puede ser informado sobre los nombres de los objetos entendidos por el juego:

```
180 OBJETOS$="ARBOLES    ARBOLES    ROC
A ...
1835 GOSUB 1890
1840 GOSUB 1885
```

```
1845 IMPRIMIDO=0
1850 FOR I=1 TO LEN(OBJETOS$) STEP 11
1855 PRINT OBJETOS$(I,I+10):IMPRIMIDO=I
MRPIMIDO+1
1860 IF IMPRIMIDO=18 THEN GOSUB 1890:IM
PRIMIDO=1:GOSUB 1885
1865 NEXT I
1885 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
4,1,0:PRINT "CONOZCO LOS OBJETOS SIGU
IENTES":RETURN
```

HELP

El desarrollo de una rutina HELP debe ser en este momento el último paso para aumentar la facilidad de manejo de nuestro juego de aventuras.

Cuando el jugador piense que se ha extraviado completamente, intentará conseguir informaciones útiles tecleando simplemente HELP.

Si se trata de un juego sencillo, porque en el transcurso del mismo se dan las indicaciones suficientes, nos facilitaremos las cosas:

```
1559 REM ************************ HELP
1560 IF ENTRADA$(1,2)<>"HEL" THEN GOTO
1600
1570 PRINT "PUEDO REPETIR LAS REGLAS DE
JUEGO.":PRINT "LO DESEA UD. ":INPUT EN
TRADA$
1571 IF ENTRADA$(1,1)="S" THEN GOSUB 80
0:GOTO 1080
1975 GOTO 1080
1979 REM ********************* FIN HELP
```


Otra respuesta frecuente es ''¡Yo lo inspeccionaría todo! - Try examining things!''.

Sin embargo, suelen darse algunos consejos, por lo menos en los juegos de aventuras recientes, que a veces no sirven de nada, porque el jugador no puede entenderlos.

La causa es que se trata de instrucciones de una sola palabra, por lo que la ayuda deseada no se refiere a un determinado objeto, circunstancia que no facilita precisamente la valoración de la situación en vistas a las dificultades que tiene el jugador.

Finalmente, el programa sólo dispone del paradero del jugador como base de trabajo. De esta forma no es nada extraño, que muchas aventuras muestren continuamente la misma información al jugador en los lugares problemáticos, y que éste no podrá entender, porque en este momento le preocupa otro problema.

Como ejemplo típico recurrimos a una escena del juego extraida del lugar 4. El jugador ha descubierto la arca del tesoro y ya lleva un buen rato intentando sin éxito alguno eliminar la cadena de hierro. En esta situación se podrá imaginar la siguiente indicación:

```
1570 IF JUGADOR=4 THEN PRINT "UN BRAZO
ALARGADO AUMENTA LA FUERZA.":GOTO 1080
```

¿Qué podría hacer un jugador, que pide ayuda en el lugar 4, con una información que le recuerda las leyes de la palanca, si ya ha fracasado en la búsqueda de la arca y, en consecuencia, todavía no puede haber descubierto la cadena?

Si queremos ayudar al jugador y acercarlo paso a paso a la

Solución en las situaciones especialmente difíciles, no podrá evitarse la formulación de extensos textos y condiciones, como ya se ha demostrado en el ejemplo precedente. Para esta misión, parece que tanto los flags como las consignas de los más variados objetos vuelvan a son los de mas utilidad.

Por otra parte, probaremos ahora esta forma de asistencia exactamente dosificada con el ejemplo de la arca, lo que no debe significar en absoluto, que esta escena se considera especialmente complicada:

```
606 M6$=''¿COMO PUEDO DESTRUIR LA CADENA
?''
607 M7$="ELLA IMPIDE LA ABERTURA DE LA
ARCA."

1970 IF JUGADOR=4 AND OB(10)=0 THEN PRI
NT "CASI ME HUBIERA CAIDO EN UN HOYO."
:GOTO 1080
1971 IF JUGADOR=4 AND OB(11)=JUGADOR AN
D NOT FL(2) THEN PRINT M6$;:PRINT M7$:
GOTO 1080
```

Recordemos, que al pasar por el lugar en cuestión, de momento sólo pueden verse árboles, y que todavía hay que descubrir el hoyo escondido entre los arbustos.

Pero si el jugador recibe información sobre una casi - caída en un hoyo, seguramente se interesará más de cerca por este obstáculo.

Cualquier otro intento con HELP no supondrá ninguna ventaja especial en este momento. Sólo cuando el arca forme parte de la decoración del lugar en forma de objeto visible, el jugador recibirá la indicación de que el examen de su contenido exige la destrucción de la cadena así como la apertura de la tapa.

Esperemos que sepa efectuar por sí sólo esta acción. Si no es así, quizás sea conveniente para este jugador reflexionar si no prefiere leer una novela o probar su suerte con juegos Arcade - Action.

Para finalizar una asistencia construída de esta forma, añadiremos una línea de seguridad, como ya la hemos conocido al programar las restantes acciones. Es una línea que se ejecutará siempre que no se haya producido una situación especialmente definida:

```
1971:
1972:
1975 PRINT "¡ PRIMERO MIRAR, DESPUES PENS
AR Y AL FINAL ACTUAR !":GOTO 1080
```

MOTIVACION DEL JUGADOR

Nuestra próxima propuesta para mejorar los juegos de aventuras ampliará nuestro motor de aventuras en una última instrucción de una sola palabra.

Es evidente que los programas de aventuras requieren algún tiempo para su solución. Pero si se juega día tras día sin apreciar ningún progreso, al final uno se pregunta si no sería mejor cargar un nuevo juego.

Por esta razón, los juegos de aventuras que trabajan con una valoración en puntos y señalan al jugador la distancia que aún le separa de la meta definitiva, parecen mucho más atractivos.

Sin embargo, hay que considerar, sobre qué base se pretende realizar la valoración. La versión habitual prevé en un

Mission - Adventure una determinada puntuación para cada tesoro que se encuentre, así como para cada misión parcial solucionada. Aparte de la indicación ''De 100 puntos has obtenido 40'',se le señala además la parte proporcional al jugador, con lo cual el estado de la partida puede ser valorado más simple y exactamente. Puede nombrarse otra vez el ejemplo del "Adventureland" de Scott Adams. Trece tesoros dan un número impar de puntos, pero con más de 50 puntos en tanto por ciento, el jugador sabe que ya ha superado más de la mitad del trayecto.

Una solución alternativa, en cambio, premiará cada paso adelante y no rehusará tampoco restar una determinada cantidad de puntos por cada ayuda prestada. Asimismo, cada error que cause la muerte será castigado, de manera que en un caso extremo incluso se puede alcanzar puntuación negativa. Pero precisamente este amplio margen es el que incita a intentar repetidamente encontrar la solución, para ser posible conseguir una mayor puntuación que el amigo, que también ha alcanzado la meta.

Si se pretende equipar de esta forma un juego de aventuras, el trabajo del programador comienza con la elección de secuencias apropiadas. Al fin y al cabo, el simple hallazgo de un objeto no debe merecer un aumento de la puntuación. Sin embargo, el jugador puede esperar con razón una recompensa, cuando vence en astucia a un monstruo o cuando encuentra un pasadizo secreto.

Antes de interesarnos detalladamente por la realización práctica de las correspondientes versiones, tendremos que conseguir previamente, que el motor de la aventura entienda la instrucción ''SCORE'':


```
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
1560
1559 REM ********************** SCORE
1560 IF ENTRADA$(1,3)<>"SCO" THEN GOTO
1600
```

A continuación introducimos la variable VALORACION para memorizar la puntuación conseguida. Para ello hay que procurar, que vuelva a ser fijada en cero con cada nuevo inicio del juego, al igual que con la muerte del personaje principal.

Sólo que la puntuación alcanzada no significa nada, si no esta expresada en relación a la puntuación máxima que se puede conseguir, por ello:

```
140 WMAX=20 :REM PUNTUACION  MAXIMA DE
LA MINIVERSION
143 VALORACION=0 :REM 0 PUNTOS
1561 PRINT"DE ";WMAX;"PUNTOS HAS OBTENI
DO ";VALORACION;"PUNTOS."
1565 GOTO 1080
```

El jugador consigue los puntos al recoger las monedas de oro y de plata. Completemos pues las correspondientes líneas en la parte de nuestra acción:

```
6014 IF N=12 THEN PRINT "O.K.":OB(12)=-
1:VA=VA+10: GOTO 1080
6017 IF N=17 AND FL(3) THEN PRINT "O.K.
":OB(17)=-1:VA=VA+10:GOTO  1080
```

Desgraciadamente, un jugador que sólo pretenda alcanzar una elevada puntuación, puede conseguir valores ilimitados con la actual realización del programa.

Recordemos, que para la formación de la acción ''Coge'' acordamos que al decir coger se refería a coger con la mano. Esto permitía en la práctica del juego, además de recoger objetos para guardarlos en el inventario, coger también objetos del inventario para realizar una acción como por ejemplo, abrir una puerta con ayuda de una llave.

Precisamente esta interpretación nos estorba ahora. Nada impide al jugador a teclear repetidas veces "Coge plata" para ir sumando cada vez diez puntos a su cuenta con ayuda de esta entrada.

Una posible solución supone modificar las condiciones preliminares generales (línea 6000), y otra significaría introducir condiciones adicionales a misma la acción.

Ya conocemos este principio por la realización de la acción ''Inspecciona''. En aquella ocasión, un arreglo parecido había que evitar, que determinado objeto volviese a ser visible infinidad de veces en su paradero inicial.

```
6014 IF N=12 AND OB(N)=4 THEN PRINT "O.
K.":OB(12)=-1:VA=VA+10:GOTO 1080
6017 IF N=17 AND OB(N)=5 AND FL(3) THEN
PRINT "O.K.":OB(17)=-1:VA=VA+10:GOTO 1
080
6019 IF N=12 AND  OB(N)=-1 THEN PRINT "
"YA TENGO LA PLATA!":GOTO 1080
6020 IF N=17 AND  OB(N)=-1 THEN PRINT "
"YA TENGO EL ORO!":GOTO 1080
```

Si usted se decide en cambio, por considerar la acción 'Coger' siempre como una acción para apropiarse de algún objeto, puede limitarse a reforzar las condiciones de entrada y a la emisión de indicaciones apropiadas:


```
6000 IF N<>JU THEN GOTO 6900
6998 PRINT ""PERO SI ESTO YA LO TENGO !
":GOTO 1080
```

Para nuestros propios juegos de aventuras, o al menos para los de este libro, procedermos a puntuar solamente los tesoros que pasan a ser propiedad del jugador. Pero no renunciaremos a puntuar cada jugada positiva.

Al contrario, aún daremos otro paso y puntuaremos a cada jugador por separado y, asimismo, le informaremos sobre la puntuación media alcanzada.

Como base de esta valoración se ofrece el número de las jugadas necesarias. Un aventurero experimentado no se conforma con inspeccionar cada menudencia, sino que además sabe sacar provecho de las indicaciones habituales.

De esta suma y de la puntuación alcanzada, determinaremos el valor medio, y a partir de este cociente procederemos a emitir el juicio definitivo del jugador.

```
146 JUGADA=0:VALORACION=0

1080 GRAPHICS 0:JUGADA=JUGADA+1

1561 PRINT "DE ";WMAX;" PUNTOS HAS OBTE
NIDO ";VALORACION;" PUNTOS":PRINT"EN "
;MOVIMIENTO;" MOVIMIENTOS."
1562 PRINT"ESTO CORRESPONDE A UNA MEDIA
DE ";VALORACION/MOVIMIENTO;" PUNTOS."
1565 GOTO 1080
```

También definimos de nuevo el criterio para un final de juego feliz. En lugar de una condición exactamente definida, que depende de cada juego, comparemos ahora la puntuación alcanzada con el máximo valor posible. Si la diferencia es igual a cero, el jugador ha resuelto la misión.

Por razones prácticas, este test se efectua al comienzo de cada jugada, antes de que se pueda transmitir cualquier tipo de datos al jugador.

```
1085 IF VALORACION=WMAX THEN GOTO 4800
```

Si además emplea la puntuación como criterio, no olvide por favor, eliminar del programa todas las restantes líneas que se ocupen de la misma tarea (números de línea entre 1331 a 1389).

Con la consecuente introducción de un sistema de puntuación, aun nos queda la tarea de adaptar el título triunfal a esta estructura. Los datos hallados deberían resumirse otra vez para que también se pueda calificar al jugador.

Sin embargo, para ello no se podrá evitar que usted efectúe una partida completa de toda la aventura, cuando haya terminado su confección y ésta ya no presente ningún fallo. En esta partida debe usted evitar todas las entradas que no sean absolutamente necesarias, y averiguar la cantidad mínima de jugadas empleada para llegar a la meta del juego. Determine la relación entre puntos y entradas necesarias, y utilice éste o un valor algo inferior como base para identificar una partida brillante.

Elija otros valores graduados correspondientemente para determinar las calificaciones más bajas.

De esta forma, la calificación de una partida de la miniversión de fiebre del oro podría efectuarse como sigue:


```
4800 GRAPHICS 0
4805 FVALORACION=VALORACION/JUGADA
4810 PRINT"  ENHORABUENA !"
4820 PRINT :PRINT :PRINT "HA RESUELTO E
L PROBLEMA PLANTEADO.":PRINT
4830 PRINT "EN ";MOVIMIENTO;" MOVIMIENT
OS HAS OBTENIDO ";FVALORACION;" PUNTOS
":PRINT"POR CADA MOVIMIENTO.":PRINT
4840 PRINT "CON ELLO HA OBTENIDO UNA CA
LIFICACION ";
4850 IF FVALORACION<0.5 THEN MO$="MISER
ABLE."
4860 IF FVALORACION>0.5 THEN MO$="REGUL
AR."
4870 IF FVALORACION>1.5 THEN MO$="MUY B
UENA."
4880 IF FVALORACION>1.0 THEN MO$="BUENA
."
4890 PRINT MO$
4899 PRINT :PRINT :END
```

Un mensaje configurado de esta forma, que además se puede adornar utilizando los caracteres gráficos, incitara a más de un aventurero que haya superado la aventura, a volver a probar su suerte con el juego.

Y todo esto sólo para ser considerado un buen jugador.

Con estas aclaraciones ya habríamos llegado al final del tema "Score", si no fuera porque aún habría que completarla con un detalle técnico, cuya falta podría desbaratar todo el planteamiento realizado hasta ahora, y disgustar seguramente a los jugadores de nuestros programas.

Quizás usted haya pensado ya en nuestra rutina Save Game, a la que también habrá que facilitar naturalmente una memorización y carga de los antes mencionados datos de valoración y movimiento, si no queremos que el jugador siempre vuelva a comenzar con cero puntos:

```
1627 PRINT #1,VALORACION
1628 PRINT #1,MOVIMIENTO

1727 INPUT #1,VALORACION
1728 INPUT #1,MOVIMIENTO
```

Todas las propuestas hechas hasta ahora en este capítulo servían sólo para aumentar la comodidad de los programas.

Creo que usted coincidiría conmigo si afirmo, que los juegos de aventuras que hayan sido confeccionados siguiendo el concepto de este libro y conteniendo todos los extras, pueden considerarse tranquilamente como juegos de clase superior. Finalicemos pues el desarrollo técnico con estas líneas, y planteemosnos la cuestión de cómo conseguir nosotros mismos que los juegos ganen en atractivo y dificultad.

Para ello procederemos como de costumbre: a las explicaciones teóricas, seguirán las correspondientes líneas de programa, referidas otra vez a nuestra primera obra fiebre del oro.

Las páginas siguientes le enseñarán especialmente, cómo puede dificultar aun más el juego al jugador, de manera que éste realmente tenga que recurrir a las ofrecidas instrucciones SAVE y LOAD para poder llegar algún día a la meta.

Para poder disponer del suficiente espacio para monstruos, trampas y demás, a continuación volvemos a la geografía para desarrollar nuestro mundo diminuto hasta convertirlo en la base de un juego de aventuras más elaborado.

Pero si tiene la intención de disfrutar usted mismo de un juego de aventuras, antes de estudiar los apartados siguientes, debería entrar todas las restantes líneas de programa, o mejor todavía, hágalas entrar por otra persona.

En el próximo capítulo podrá encontrar el correspondiente listado, que incluye todas las propuestas realizadas.

ORIENTACION Y DESORIENTACION

Ya se mencionó al principio del libro, que la mayoría de juegos de aventuras se desarrollan en un mundo que conste de unos cuarenta lugares. Sin embargo, aparenta ser un mundo mucho más grande, como hemos visto con el inmenso bosque, un pantano, unas cuevas subterráneas, un laberinto y...

Pero en realidad, estas zonas que parecen ser tan amplias, son grupos de unos pocos lugares comunicados de manera tan astuta, que el jugador normalmente no se da cuenta por qué camino ha llegado a ellos.

Para ello se dispone de distintas posibilidades que van desde el simple lugar donde ocurre todo, hasta una sucesión de lugares, a cuyo término el jugador vuelve para empezar un nuevo recorrido.

Aprovechemos en esta ocasión los lugares 1 y 2 de fiebre del oro, para disponer ya al principio del juego de una vasta superficie.

Haga que el lugar 1 desemboque en si mismo en las direcciones norte y oeste, proceda igualmente con lugar 2, e intente imaginarse después al jugador, que comienza en lugar 1 y camina hacia el oeste. ¿Comó puede éste saber, que después de cada entrada que haya efectuado aún se encuentra en el mismo lugar?

Si no tiene la suerte de empezar la partida efectuando la entrada "E", necesitará realizar una buena cantidad de entradas para encontrar el camino que le saque del bosque, cosa que ya le impide conseguir una buena puntuación.

```
501 DATA 1,1,1,2,0,0
502 DATA 2,2,3,1,0,0
```

La búsqueda dentro del camino correcto resultará todavía más difícil, si la salida sur de lugar 2 no permite que el jugador vuelva a entrar en el lugar 2, sino en el lugar 1.

```
501 DATA 1,1,1,2,0,0,
502 DATA 2,1,1,3,0,0
```

Al iniciar el programa se sorprenderá del efecto que causa esta pequeña modificación.

De esta forma ya se pueden construir pequeños laberintos con sólo tres lugares, al efectuar las comunicaciones sin orden ni concierto, sin ninguna relación con la brújula, existiendo una única entrada o salida.

La técnica habitual que también nosotros hemos empleado para comunicar los lugares, se guía por los caminos reales, fijados por los puntos cardinales. Con ello se origina un mundo que permite identificar sin problemas del lugar y facilita la orientación, sobre todo si el jugador confecciona un mapa apropiado.

Sin embargo, este idea vuelve a ser inútil cuando se puedan saltar varios lugares.

Imaginemos tres lugares consecutivos, que todos empiecen con la descripción ''Estoy en una cueva inmensa''.

Un jugador que entre en la cueva 1 por el oeste, al no ver nada especial, querrá en principio conocer los alrededores a grandes rasgos, y probablemente no cambiará la dirección una vez tomada, porque necesitará encontrar fácilmente el camino de vuelta.

Después de atravesar los lugares 2 y 3, el jugador vuelve a aparecer en el primero, cosa que no será capaz de averiguar sin realizar las correspondientes acciones.

La utilización de esta técnica de salto permite construir fácilmente laberintos que parecen ser inmensos. Hay suficiente con seis lugares para hacer casi imposible que el jugador encuentre el camino correcto.

Incluso el propio programador tendrá bastantes dificultades al realizar la fase de prueba.

Esta forma de trabajo dificulta incluso la confección de un mapa. Sin embargo, no ocurre lo mismo al utilizar el procedimiento alternativo que aquí presentamos.

La forma de proceder según este método parece exigir, la confección del laberinto en papel preferiblemente cuadriculado, tal como también pueden encontrarse frecuentemente en los pasatiempos de diversas revistas.

Al realizar la programación, cada cuadrícula corresponde a un lugar, lo que ya permite adivinar el considerable trabajo que esto representa.

También es comprensible, que durante su excursión el jugador coja una nueva cuadrícula con cada nuevo lugar, marque las paredes y deje los pasillos en blanco.

En su lugar, coja usted seis lugares de los cuales fijará a uno como lugar de entrada, a otro como lugar colectivo y a un tercero como lugar de salida. Es importante tener en cuenta, que el lugar de entrada conduzca exclusivamente y en una única dirección de vuelta hacia el último lugar visitado.

Lo mismo vale para el último lugar. Solamente un único pasillo permite el acceso al nuevo mundo, mientras que todos los demás caminos deben conducir de vuelta hacia el laberinto.

El lugar colectivo tiene una función central. En cada lugar del laberinto hay dos o más caminos que conducen hacia este lugar central, mientras que todas sus salidas vuelven a llevar al jugador al primer o segundo lugar del laberinto.

Para complicar aún más el asunto, se disponen las salidas del lugar, que se encuentra antes del último y en comunicación con éste, de al manera, que sólo exista una única salida que conduzca hacia el último lugar, mientras que todas las demás lleven hacia el lugar colectivo o al primero.

No cabe mencionar, que se prevé una descripción idéntica para todos los lugares y que se utilizan las seis direcciones si esto es posible.

Después de todo esto, el jugador necesitará casi tanta suerte como en la lotería para poder salir del laberinto. Lo pasará mal antes de encontrar el camino correcto en el penúltimo lugar, lo mismo que en el último. Cada error lo devuelve además, hacia otro lugar que, por su parte, también ofrece seis posibilidades de movimiento.

Intente aclararse en el siguiente laberinto. Aunque usted mismo haya acabado de entrar las correspondientes conexiones, tendrá sus dificultades en resolverlo:

El laberinto de fiebre del oro:

```
110 AR=14: ...
207 PRINT "EN LA BOCA DE ENTRADA DE UNA
 VIEJA MINA.":RETURN
219 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
224 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
225 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
226 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
227 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
228 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN
229 PRINT "EN UN FILON SERPENTEADO.":RE
 TURN


507 DATA 7,0,1,5,0,0
519 DATA 8,0,0,5,0,0
524 DATA 10,7,8,8,8,8
525 DATA 11,8,8,13,11,8
526 DATA 11,8,10,11,11,8
527 DATA 9,8,10,11,10,8
528 DATA 8,10,10,11,0,0
529 DATA 0,11,12,11,0,0
```

CAMBIO DEL LUGAR

Además de laberintos de todos los tamaños, también disponemos de otros medios para dificultar la orientación del jugador.

Gozan de mucha popularidad las súbitas transferencias del personaje principal hacia otro lugar, hecho que se comunica al jugador con el mensaje "Todo a mi alrededor da vueltas".

Pero este mensaje puede considerarse un gesto amable del programador, y no una obligación.

Los causantes de este tipo de acciones suelen ser los objetos mágicos que se activan frotando fuertemente, al igual que una lámpara de Aladino.

Pero también conjuros y brebajes mágicos producen sus efectos, por lo que se aconseja al jugador de un programa de aventuras, que se ande con el mayor cuidado si aparece la palabra 'magia' en relación con cualquier cosa.

Desde el punto de vista de la técnica de programación, estos suubitos cambios de lugar no comportan ningún tipo de dificultad en nuestro sistema de aventuras, ya que basta con asignar el número del nuevo lugar a la variable JUGADOR.

La orden ''Frota lámpara'' podría implementarse de la forma siguiente (lámpara = objeto 3):

```
xxxx IF N=3 AND OB(3)=-1 THEN PRINT "EL
MUNDO DA VUELTAS ...":JUGADOR=9:GOTO 1
080
```

Si, además, la descripción del lugar 9 corresponde a la del anterior lugar, no podrá evitarse, que el jugador este completamente confundido después de la próxima jugada. Esto ocurre especialmente en el caso, de suprimir el

Mensaje o se escojer otro texto:

```
xxxx IF N=3 AND OB(3)=-1 THEN PRINT "O.
K. - NO PASA NADA.":JUGADOR=9: GOTO 10
80
```

Se puede superar el grado de dificultad, cuando el cambio de lugar del jugador ya no dependa ni de las manipulaciones de algún objeto, sino siendo controlado única y exclusivamente por la casualidad.

La estructura de nuestro programa de aventuras nos permite prescribir, que el hecho de atravesar un lugar determinado sea una acción inevitable.

Si el contenido de JUGADOR coincide con el número de este lugar, se toma una determinación casual que decide, si el jugador es trasladado o si se le permite una estancia en el lugar actual para poder realizar otras acciones.

¿Qué jugador se arriesgaría a repetir el proceso, cuando haya sido devuelto repetidas veces al laberinto después de atravesarlo felizmente y entrar a continuación en un lugar aparentemente inofensivo?

Un fragmento de la aventura fiebre del oro vuelve a mostrar la forma en que se inserta este obstáculo:

```
110 AR=16: ...

221 PRINT "EN UNA GALERIA..":RETURN
222 PRINT "EN UNA ........

519 DATA 8,0,15,5,0,0
521 DATA 11,11,11,7,11,0
522 DATA 0,15,0,0,0,0

1345 IF JUGADOR=15 THEN IF RND(1)>.7 TH
```

EN JUGADOR=16

El traslado del jugador hacia el lugar crítico se realiza al abandonar el lugar anterior al laberinto - actualmente el lugar 8, un filón serpenteado - en dirección oeste.

El jugador puede retirarse inmediatamente de este lugar, pero también puede intentar averiguar lo que le tiene preparado el contorno.

Si la casualidad quiere - y el hecho de que quiera se determina con el valor comparativo en la función RND -, se detiene en el siguiente lugar. En caso contrario, se encontrará otra vez en un lugar del laberinto (lugar colectivo 11), que no le ofrece ninguna referencia para orientarse.

Si no le acompañó la suerte, después de unos cuantos intentos el jugador decidirá, que se trata de otro acceso mas al laberinto y evitará este lugar a partir de ahora, con lo cual no podrá encontrar nunca una parte del tesoro.

ACCESOS OCULTOS

Los accesos ocultos le pueden causar las mismas dificultades al jugador. Incluso aunque en la línea ''Puedo ir hacia ...'' sólo se indiquen las direcciones este y oeste, esto no significa que el mundo acabe en el norte:

```
VEO UNA CUEVA SOMBRIA, UN OSO DE MIRADA
FEROZ, *PEPITAS DE ORO*.

PUEDO IR HACIA OESTE, ESTE.
```



```
O.K.

¿QUE DEBO HACER ?N
¡HACIA ALLI NO CONDUCE NINGUN CAMINO!

¿QUE DEBO HACER ?E
```

Muchos jugadores - cuando se trata de aficionados, incluso la mayoría - se fiarán de las indicaciones realizadas y no entrarán nunca en la cueva. Esto se debe al hecho de permitir, que se efectúe una inspección - así como otras acciones - desde la posición actual, disipando de esta forma la sospecha de que allí se pueda esconder un inmenso mundo subterráneo.

Sólo una entrada como "Entra cueva" nos lleva a hacer nuevos descubrimientos:

```
170 VERBOS$="INSPECCIONA... ...ENTRA "

2200 ON VN GOTO 5000,6000,7000,8000,900
0,10000,13000

13010 IF N=13 AND JUGADOR=5 THEN JUGADO
R=12: PRINT "O.K.":GOTO 1080
```

ATENCION: Antes de comprobar la función de esta líneas, es imprescindible entrar también todos los demás lugares, así como modificar las conexiones presentes hasta ahora (líneas 200 - 300 , 500 - 600).
Asimismo hay que adaptar la línea ||10 a la nueva situación.

Lo mejor de esta realización es que ni siquiera se camufla el acceso - en contraposición a otra clase de realización -, razón por la cual también goza de gran popularidad entre los productores de aventuras.

Ahora la misión del jugador ya no se limita sólo en descubrir un lugar oculto y conseguir entrar en él con ayuda de palabras correctas, sino que deberá ser capaz de desarrollar un plan sobre la forma de construir un camino transitable hacia este lugar.

En el caso más simple, se trata de una puerta cerrada. Una apropiada indicación realizada por el motor se garantiza tanto a través de las descripciones de objetos, así como de los datos de la línea responsable de este lugar y que pertenece a la tabla de direcciones:

```
180 OBJETOS$="...PUERTA"
3xx T$="UNA PUERTA":RETURN
410 DATA.,.,...,0

REM LUGAR 1
2xx PRINT "EN UNA CELDA.":RETURN
5xx DATA 0,0,0,0,0,0
```

Este habitual procedimiento prevé que el jugador descubra la puerta provista de un candado macizo, al inspeccionar la celda.

Las órdenes "Abre puerta" o "Abre candado", señalan la ausencia de llave, provocando de esta forma, que el jugador compruebe todos los restantes objetos que forman parte del lugar por su utilidad para poder abrir el candado.

En la mayoría de aventuras, basta con que la llave este presente en el inventario, mientras que los demás exigen además, una utilización apropiada de la llave,

característica que ahora podemos entender gracias a nuestra experiencia práctica.

Mientras que los primeros se conforman con una simple comprobación (OB(N)=-1), los demás programas - más elaborados - trabajan adicionalmente con flags.

Sea como sea, una vez abierta la puerta, aparecerá una determinada indicación de dirección en una línea de encabezamiento, que nos permitira deducir la manipulación de la línea responsable de este lugar del campo PASADA (,).

```
xxxx IF N=y AND OB(z)=-1 AND JUGADOR=r
THEN PRINT "O.K.": PASADA(r,ri)=nr: GO
TO 1080

y = número de objeto puerta
z = número de objeto llave
r = Lugar actual
ri= punto cardinal correspondiente
nr= nuevo lugar a alcanzar
```

La línea precedente, insertada en la parte de programa prevista para el tratamiento del verbo "Abre", se asegura de que el jugador esté en posesion de la llave (z) correcta, así como delante de la puerta (r) correcta. A continuación, se modifican adecuadamente las indicaciones de la tabla de direcciones.

LIMITACIONES Y CONTADOR

Pienso que son elementos especialmente interesantes para que un juego de aventuras pueda destacar sobre la gran masa. Principalmente son ellos los que proporcionan la fascinación de la realidad a los programas de aventuras, y configurar

Algunas secciones de estos juegos en verdaderas rompecabezas.

Por esta razón, su campo de acción no se encuentra tanto en una activa participación en el juego, como en un control de fondo de las condiciones del margen que rodean el juego.

Una vez llegado a este punto hay que mencionar la limitación del inventario, ya que un límite de carga forma parte ya de la completa elaboración de un juego de ordenador.

Esto es comprensible. En primer lugar resulta increíble, que una sola persona - partimos del hecho de que no se trata de una aventura de superman - pueda llevar consigo infinidad de objetos, y en segundo lugar, el juego puede resultar demasiado fácil cuando siempre se puede disponer inmediatamente de todos los objetos.

Desde es punto de vista técnico, un contador comprueba el número de objetos que el jugador lleva consigo. Si se supera la fuerza de carga del personaje principal, prevista por el programador, se rechaza la recogida de mas objetos.

Esto exige la definición del máximo número de piezas admitido antes de comenzar cualquier acción.

```
110 ... IMAX=5
```

Por lo demás dejamos la rutina de inventario tal como está, ya que debemos dedicar mayor atención al proceso de coger.

Ahora se exigirá otra condición común a todas las acciones que utilicen el verbo "Coge"; por ello conviene modificar por segunda vez la línea 6000.

Desgraciadamente, esta planificación no prevé sólo una modificación de la estructura del bloque afectado, sino que además echa a perder nuestro sueño de un motor universalmente aplicable con todas sus funciones y restricciones.

Por este motivo modificamos la tabla de saltos en la línea 2200 y efectuamos la prueba necesaria antes de la propia ejecución de las instrucciones:

```
2200 ON VN GOTO 5000,2210,7000,8000,900
0,10000
2209 REM ******** TEST POR SITIO EN INV
2210 CANTIDAD=0
2220 FOR I=1 TO AO
2230 IF OB(I)=-1 THEN CANTIDAD=CANTIDAD
+1
2240 IF CANTIDAD=IMAX THEN PRINT ''NO GR
ACIAS. - ¡ YA LLEVO BASTANTE CARGA !'':
GOTO 1080
2250 NEXT I
2260 GOTO 6000 : REM *** EJECUCION COGE
2270 REM ****************** FIN INVTEST
```

Mientras que al contador le toca un papel pasivo, y la rutina sólo será llamada dentro de una acción muy definida, otros controles de fondo no sólo limitan al jugador en su libertad de movimiento, sino que incluso pueden provocar el final de la partida.

Esta afirmación es especialmente válida para las siguientes líneas de programa, cuya inclusión en nuestro sistema representa el punto sobre la "i".

AVAILABLE LIGHT

Esta es la expresión usual con que se designa una propiedad que puede encontrarse en algunos programas de aventuras, y que puede llevar al jugador a la desesperación mediante obstáculos adicionales.

Parece que los jugadores de aventuras apasionados no tienen suficiente con los obstáculos habituales y, como ya sabemos, no hay nada más difícil que la vida cotiana, sólo hay que imitar precisamente esta vida de forma más perfecta, para remediar el aburrimiento de un juego demasiado fácil.

Sin "luz disponible" estamos a oscuras y difícilmente podremos ver alguna cosa, y mucho menos actuar con precisión. Si a todo ello se añade el hecho de que no nos hallamos en nuestro dormitorio, donde este hecho no tendría tanta importancia, sino en un paisaje rocoso de naturaleza salvaje y escabrosa, cada paso se convierte en un riesgo mortal.

Esto es muy oportuno para nosotros, pues incluyendo las siguientes líneas en nuestro juego de aventuras, imaginemos la reacción de nuestros amigos, a los que naturalmente presentaremos nuestra obra una vez terminada.

Afortunadamente, cualquier programa de aventuras se presta a aceptar este obstáculo. A menudo disponemos de vastas cuevas o sistemas de pasadizos subterráneos, mientras que otros programas se desarrollan en el interior de una casa. Todos ellos son lugares que requieren una iluminación artificial. Si alguna vez se presentase la situación de que toda la acción ocurra al aire libre, habrá que imitar el movimiento del sol.

Para poder realizar una salida y puesta de sol, sólo hace

falta introducir otro contador.

Para empezar, fijamos la duración del día (de la noche), sirviéndonos para ello de las entradas que efectúe el jugador como unidad de tiempo.

Supongamos, que después de cada 50 jugadas tras la salida de sol emplece la noche. Este valor será igualmente propio de cada juego de aventuras como la cantidad de puntos alcanzable, por lo cual se inicializa la correspondiente variable LM ya al comienzo del programa:

```
110 ... IMAX=5:LM=50:LLUZ=-1
```

Sus ya considerables conocimientos sobre el tema aventuras le permite considerar la variable LLUZ como un flag, en el sentido que un -1 significa una cantidad de luz suficiente para realizar todas las acciones, mientras que 0 significa oscuridad.

Las dos posiciones posibles de este interruptor indican a la rutina de salida de nuestro motor, si el jugador es capaz de ver alguna cosa, o si debe andar a oscuras:

```
1091 REM ********************** SIN LUZ
1092 IF LLUZ THEN 1100
1093 PRINT "NO SE EXACTAMENTE DONDE EST
OY."
1094 PRINT " ES DEMASIADO OSCURO PARA V
ER ALGO."
1095 PRINT: PRINT"TAMPOCO PUEDO VER LAS
BOCAS DE SALIDA.":GOTO 1320
```

Después de la línea 1092 se procesarán las rutinas de salida no utilizadas hasta ahora, en el caso de que exista luz.

Si ocurre lo contrario, las líneas 1093 a 1095 informan en

la forma habitual de que no hay nada que ver.

Al imprimir la línea de separación (1320), se puede continuar la ejecución del programa a través de las líneas habituales.

Ya sólo nos queda procurar la correcta posición
caso, que debe realizarse después de efectuar el número de jugadas fijado con LM.

Para realizar esta operación de comparación no se puede consultar el contador de jugadas, porque al cambiar de día a noche también se exige, que el contador se ponga otra vez a cero. Por ello introducimos la variable cambio de luz (LW):

```
110 ... IMAX=5:LM=50:LLUZ=-1:LW=0

1080 ...MOVIMIENTO=MOVIMIENTO+1:LW=LW+1
1084 IF LW=LM THEN GOSUB 3000


2999 REM ** SUBPROGRAMA INTERRUPTOR LUZ
3000 IF LLUZ=-1 THEN LLUZ=0:GOTO 3020
3010 IF LLUZ=0 THEN LLUZ=-1
3020 LW=2
3030 RETURN
```

Al principio de una nueva partida de entradas, se comprueba si se ha alcanzado la duración (LM) prevista de un período. Si el resultado de la prueba es afirmativo, se llama al subprograma interruptor y se modifica el estado de LLUZ. No hace falta nada más para exponer al aventurero a un cambio de día y de noche, cosa que puede significar que éste busque una lámpara.

Sin embargo, no queremos limitarnos a simular el movimiento

del sol, sino que hacemos depender la acción de la posesión de una lámpara.

Con ello encargamos al jugador la misión adicional de disponer siempre de suficiente combustible para su fuente luminosa.

En nuestro caso se tratará de aceite, que puede hallar en la cueva del oso. En esta cueva llena de aceite las botellas que debe haberse llevado previamente y, a continuación, la lámpara.

Cada llenado vuelve a poner el contador cambio de luz al valor máximo.

Para complicar el asunto adicionalmente, la lámpara todavía contendrá un resto de aceite en el momento en que el jugador la encuentre. Si después entra en la mina sin rellenar la lámpara, estará perdido sin posibilidad de salvación. Otro paso más y el aventurero tropieza desnucandose.

Ya que no hemos previsto oscuridad para el resto del territorio, el primer paso para realizar esta versión consiste en modificar los valores iniciales.

Una conmutación siempre tiene lugar, cuando el contador cambio de luz coincide con el el contenido de LM (cantidad de luz).

Sin embargo, para permitir que el jugador pueda realizar un número casi ilimitado de jugadas, fijamos LM en cero y LW en uno, y después aseguramos que LW no pueda ser nunca igual a LM:

```
110 ... IMAX=5:LM=0:LLUZ=-1:LW=1:LW=1:L1
=20
```

Se añade L1 porque el contenido todavía existente debe ser menor que el posteriormente añadido. El resto de las líneas pueden tomarse del ejemplo anterior.

Esta rutina no se activa hasta que se efectua el primer encendido de la lámpara. Además de emitir "O.K.", la ejecución de esta instrucción también inicializará correctamente los contadores:

```
11030 IF N=23 AND OB(23)=-1 THEN PRINT"
O.K. - LA LAMPARA ESTA ENCENDIDA.":LM=
L1:LW=1:GOTO 1080
```

El jugador dispone ahora de luz suficiente para realizar las veinte acciones siguientes y, rellenando la lámpara puede aumentar este valor hasta cincuenta:

```
12010 IF N=23 AND OB(23)=-1 AND FL(6) T
HEN L1=50:LM=50:LW=1:PRINT "O.K.":GOTO
1080
```

Para permitir el rellenado de combustible con llama sea encendida o apagada, asignamos a la variable L1, así como a la LM, la cantidad de jugadas a efectuar ahora.

En el caso de que el jugador se dirija hacia la superficie o apague la lámpara por alguna otra razón, debemos salvar el actual contenido de combustible.

Primeramente, hay que añadir la palabra "Apaga" a VERBOS$, y prever en la línea 2200 un destino de salto para el tratamiento de esta acción.

```
14000 IF N=23 AND OB(N)=-1 THEN L1=LM-L
W:PRINT "O.K.":GOTO 1080
```

Naturalmente también hay que salvar estos valores al grabar un juego:

```
1625 PRINT #2,JUGADOR:PRINT #2,L1:PRINT
#2,LM:PRINT #2,LW
1725 INPUT #2,JUGADOR:INPUT #2,L1:INPU
T #2,LM:INPUT #2,LW
```


Ahora ya nos hemos asegurado, que el asunto de la luz funcione realmente en la forma prevista.

Sin embargo, para evitarle al jugador la súbita sorpresa del final irremediable - porque si se encontrase en las profundidades de la tierra, cuando se le acabase el combustible, ya no podrá salvar su pellejo - debería emitirse un aviso para jugar limpio.

La siguiente línea le informará, un poco antes de apagarse la lámpara, del peligro que corre:

```
1350 IF LM-LW<15 THEN PRINT "LA LUZ DE
LA LAMPARA SE EXTINGUE."
```

Espero, que los ejemplos precedentes le sirvan de ayuda para poder realizar sus propias ideas.

Así, no le será difícil encargarse en sus programas, por ejemplo, de que el jugador se alimente a tiempo, para poder continuar cogiendo objetos en sus manos y realizando acciones.

Del mismo modo, con ayuda de otro contador, puede hacer que las reacciones dependan del tiempo transcurrido.

CASUALIDADES

Terremotos que provocan desprendimiento de rocas, monstruos que aparecen súbitamente, los enemigos naturales del protagonista en el juego de aventuras, así como trampas que se abren en el momento más inoportuno, son también otros métodos populares para apartarlo del camino correcto.

Normalmente, la aparición de estos sucesos se controla por la casualidad, siendo incluso en doble sentido.

Un monstruo puede habitar un determinado lugar y arrancarle la cabeza al jugador:

```
1360 IF JUGADOR=10 THEN MO$="ESTA VEZ N
O TENIA MIEL PARA EL OSO.":GOTO 4500
```

Pero igualmente, la trampa puede actuar sólo en algunos casos, mientras que en otros momentos le puede perdonar la vida al aventurero:

```
1370 IF JUGADOR=20 AND RND(1)>.8 THEN M
O$="EL SUELO SE HUNDIO BAJO MIS PIES."
GOTO 4500
```

La tercera versión combina todos los elementos presentados hasta este momento.

Asimismo se puede partir, de la movilidad de los enemigos del protagonista, razón por la cual el monstruo también debería disponer de una determinada sección del mundo de aventuras para poder actuar:

```
1380 IF (JUGADOR=13 OR JUGADOR=14 OR JU
GADOR=16) AND RND(1)>.8 THEN MO$="OTRA
VEZ EL OSO.":GOTO 4500
```


DE LA AVENTURA EN TEXTO A LA GRAFICA

Los programas de aventuras presentados recientemente suelen mostrar una excelente presentación óptica además de las características ya mencionadas. Existe una imagen distinta para cada lugar, que informa al jugador con más o menos precisión, sobre cosas importantes para el juego y sobre todo lo que los rodea.

Para realizar estas aventuras gráficas desde el punto de vista de la técnica de programación, el programador dispone de tres posibilidades.

En principio puede grabar en un diskette una gráfica para cada lugar y cada objeto y cargarlas en la memoria de pantalla del ordenador cuando sea necesario. En la actualidad, este método se utiliza en prácticamente todas las aventuras gráficas, por un lado, porque pueden conseguirse resultados magníficos mediante digitalizador de video u otros medios auxiliares, como es el koalapad. Por otro lado, el trabajo adicional de programación consiste en un único subprograma que debe cargar un fichero en función del número de lugar (JUGADOR), y puede constar de una sola línea en el caso más favorable.

Naturalmente se requiere el espacio correspondiente en el diskette (ahora entenderá por qué algunas aventuras ocupan tres o más diskettes), y el jugador tendrá que soportar con paciencia el tiempo de carga.

El segundo método trabaja con un interpretador gráfico, una parte del programa de aventuras que dispone de una instrucción line y paint, y al que hay que proporcionar cada vez los datos de todos los contornos y colores.

Este método es el que utiliza menos memoria , ya que generalmente se necesita sólo un byte para cada punto y,

Si se ejecutan las líneas en grupos, basta sólo una posición de memoria para memorizarlas.

Por lo demás, el programador gráfico que presentamos a continuación también trabaja según el mismo principio cuando, al pulsar CTRL Z, muestra una nueva imagen.

Finalmente, el tercer procedimiento destaca por su rapidez, porque cada instrucción necesaria para construir la gráfica es ejecutada directamente. Por supuesto es necesario, que el sistema operativo del ordenador disponga de estas instrucciones y que la memoria de trabajo pueda admitir las órdenes necesarias.

Al seguir desarrollando nuestra aventura, nos limitaremos a los dos últimos métodos, ya que para su aplicación no precisamos ni medios auxiliares técnicos ni laboriosos subprogramas.

Ambas soluciones requieren naturalmente pequeñas modificaciones en nuestro programa motriz, puesto que ahora las salidas deberán efectuarse en otra parte de la pantalla.

Vuelva a cargar pues la miniversión de fiebre del oro y cambie GRAPHICS 0 (línea 1000) por GRAPHICS 8.

Después de encender el modo operativo gráfico, ya sólo disponemos de cuatro líneas de texto en nuestra imagen del monitor. Esto significa de entrada, que debemos limitar nuestras emisiones; sólo se permite la cantidad realmente imprescindible.

Podemos prescindir de la línea vacía y del guión, por lo que borramos primero las líneas 1310 y 1320. Tampoco nos hace falta ya la instrucción de posición en 1390, ya que lo más seguro es que. lleguemos de todos modos a la última línea.

Si ahora ejecutamos el programa, las diez líneas vacías en 1090 ya carecen de sentido, por lo que también podemos suprimirlas.

Puesto que a pesar de ello seguimos utilizando más de cuatro líneas de pantalla, prescindiremos también de la descripción del lugar. Si nos esforzamos mínimamente en confeccionar la grática, esta descripción sobrará.

Borre pues la línea 1100, así como todas las descripciones de lugar dentro de las líneas 200 a 299 del programa.

Nuestro próximo paso consiste en resumir los mennsajes sobre los caminos disponibles y sobre la invitación al jugador para indicar sus instrucciones.

Para no tener que utilizar más de una línea de pantalla, modificaremos las líneas 1011 a 1016 de tal manera, que ya solo sea preciso escribir la letra inicial, la coma, y un espacio en T$. A continuación complete la línea 1310 con un punto y coma, para evitar el avance de línea, y borre también la 1260, ya que tampoco se llegará nunca al final de la línea, aunque se indiquen las seis direcciones.

Después de iniciar el programa, debería recibir la siguiente imagen:

VEO MUCHOS ARBOLES GRANDES.
E.?

```
O.K.
VEO MUCHOS ARBOLES GRANDES,
ALGUNOS PEDRUSCOS.
O, E.?
```

Según esto, se muestran ahora en la línea superior las informaciones, seguidas de las descripciones de los objetos. Para poderlas emitir, el juego de aventuras dispone de dos líneas de pantalla; una tercera línea se encarga del scroll de las informaciones visualizadas. Procure pues, elegir descripciones lo más escuetas posible al confeccionar su propio juego de aventuras gráficas. Esta solución está perfectamente justificada, puesto que el jugador prestará más atención a la imagen que al texto.

Nuestra miniversión de la fiebre del oro también deberá editarla siguiendo el mismo procedimiento.

ESTRUCTURA DE LA IMAGEN

Es preferible, encargar la estructura de cada gráfica a algunos subprogramas, que serán llamados por el motor.

Hay que tener en cuenta que esta rutina de visualización no precisa ser reclamada después de cada entrada efectuada por el jugador, puesto que no es necesario dibujar de nuevo el mismo lugar después de cada acción realizada en él.

Por esta razón introducimos otra variable de control:

```
1090 IF VIEJO<>JUGADOR THEN GOSUB 30000
1100 VIEJO=JUGADOR
```

Si hay que representar un lugar en la pantalla, nos

dirigiremos al bloque que contiene las rutinas gráficas.

A continuación, se transmite su número a la variable VIEJO. Si el jugador se aleja entonces del lugar de los hechos, se cumple la condición de la línea 1090 y se vuelve a llamar a la 30000, que escogerá el número de caracter correcto en función del número de lugar:

```
30000 ON JUGADOR GOSUB  31000,32000,330
00
30010 RETURN

31000 REM DIBUJAR LUGAR 1
31999 RETURN

32000 REM DIBUJAR LUGAR 2
31999 RETURN
```

Después siguiendo la ejecución del programa se emite la descripción de los objetos en la forma habitual.

Si usted prefiere ahora, construir las gráficas con ayuda de un interpretador, inserte a partir de la línea 30000 las correspondientes rutinas del programado. gráfico, que le presentaremos más adelante.

Para ello sólo necesita las dos modalidades cargar imagen y dibujar imagen, que comienzan en las líneas 7000 y 5000, respectivamente.

CAPITULO 5
- PRACTICA DE AVENTURAS -

Este capítulo no le enseñará como hacer las cosas, sino cuáles puede hacer, mediante el ejemplo de programas enteros.

A continuación se presentamos los listados completos de tres juegos de aventuras, confeccionados todos ellos a partir de las partes de programa y rutinas desarrolladas en este libro.

El primero de estos programas es el fiebre del oro, una aventura que ya se ha comentado ampliamente; en segundo lugar, debe mencionarse el "Castillo encantado". Sin embargo, la culminación es el "Space Mission", un juego lleno de trucos y confecionado como aventura gráfica.

Desgraciadamente ocurre que, después de analizar este libro, usted podrá considerar los programas prácticamente como instrucciones para llegar a la solución. Por ello, se aconseja compartir el trabajo del tecleado con otra persona.

Nos referimos, sobre todo, a partir de la línea 5000, donde se ha programado la acción del juego.

Seguidamente, le presentaremos los prometidos generadores de programas, tanto el programador gráfico como el Venturefix.

la unión de ambos programas le permitirá crear juegos de aventuras controlados por menú, e incluso podrá completarlos con un mínimo esfuerzo hasta obtener impresionantes programas gráficos.

Las indicaciones e instrucciones de uso precisas se darán antes de cada programa.

INSTRUCCIONES DE JUEGO: AVENTURAS

Una vez cargado el programa en cuestión y ejecutarlo con RUN, después de la imagen titular y de otras indicaciones generales sobre el juego, aparecerán las primeras descripciones e informaciones en la pantalla.

Usted debe distinguir entre la mitad superior e inferior de la pantalla: - en la superior se describe el lugar donde usted se encuentra en un momento determinado, indicándole todos los objetos visibles. En otra línea de la pantalla se hace referencia a todas aquellas direcciones a las que usted se puede dirigir.

La mitad inferior, en cambio, le sirve para indicar sus instrucciones al protagonista del juego. Asimismo, en esta parte se muestran las informaciones que dependen de la ejecución de sus entradas efectuadas.

SUS ENTRADAS

Sus entradas habituales constan normalmente de dos palabras, un verbo y un objeto. Compruebe al empezar la longitud de entrada exigida por la aventura en cuestión. Casi siempre bastan las primeras tres o cuatro letras para identificar una palabra.

Procure emplear los mismos términos que se utilizan en las descripciones.

Sin embargo, esto no significa que deba reaccionar a un mensaje como:

VEO UNA PUERTA ROJA.

QUE DEBO HACER:

con ABRE PUERTA ROJA. Normalmente basta con utilizar ABRE PUERTA.

También se admiten otro tipo de entradas llamadas instrucciones de una sola palabra, que no actúan directamente sobre la marcha del juego, sino que ofrecen determinados servicios complementarios a modo de información.

TRASLACION:

Generalmente, usted puede moverse en las direcciones indicadas. Para ello sólo tiene que teclear la letra inicial de la correspondiente dirección (excepto al moverse hacia arriba o hacia abajo, para lo cual usará AR o AB): Norte, Sur, Este y Oeste: N, S, E, O; arriba y abajo: AR, AB.

Sin embargo, también hay excepciones, y una entrada como ENTRA ... puede sernos de gran utilidad.

MANIPULACION DE OBJETOS

Naturalmente, usted no querrá limitarse sólo a viajar a través del mundo imaginario, sino que deberá entrar en acción y cumplir alguna misión.

Esto presupone, que lo primero que tiene que hacer es descubrir los objetos propios de ese mundo. A continuación, los podrá INSPECCIONAR, COGER, UTILIZAR, etc. (si alguna vez no sabe cómo continuar, debería acordarse de estas palabras).

Para alcanzar de la forma más rápida posible su objetivo, deberá proceder a partir de este momento de forma lógica y ordenada.

Supongamos, que usted encuentre una llave en algún sitio. Seguramente pensará ¡estupendo!, pero ¿qué puedo hacer con ella?

En la vida real, usted se fijaría bien en la llave, la inspeccionaría y después decidiría si dejarla donde estaba, o si se la llevaría por considerarla útil o incluso valiosa. También tendrá que realizar cada uno de estos pasos en la aventura.

Entre: INSPECCIONA LLAVE. Quizás obtenga la respuesta: ES LA LLAVE DE MI PISO o: ES UNA LLAVE DE COCHE o: ES DE ORO PURO. Naturalmente, usted no se dejaría ninguna de estas llaves, sino que se las llevaría todas, efectuando pues la entrada: COGE LLAVE. En la pantalla se muestra al menos el mensaje O.K. (podrá ver este O.K. a menudo; significa, que el interpretador de la aventura ha entendido y ejecutado su entrada), y también pueden seguirle otras reacciones.

Naturalmente, ahora la llave ya no aparecerá en la línea superior VEO, puesto que por habersela llevado, ya se encuentra en el bolsillo de su abrigo o algo parecido, pero ya no está en el lugar.

Si se hubiesen acumulado de esta forma varios objetos en sus bolsillos, probablemente usted perdería el control de todos. Por esta razón se le ofrece la posibilidad de hacer un

INVENTARIO

Siempre que usted desee saber, los objetos inútiles que lleva consigo, o averiguar que fue lo qué le hizo hundirse en las arenas movedizas, o bien determinar que objetos le pueden servir para resolver cierto problema, entonces sólo tendrá que teclear INV.

La respuesta inmediata será: LLEVO CONMIGO LO SIGUIENTE:, seguida de una lista de todos estos objetos.

INTERRUMPIR LA PARTIDA

Seguramente será incapaz de resolver una aventura con un sólo intento, y naturalmente no querrá volver a empezar de nuevo al día siguiente.

Por ello casi todos los juegos de este tipo permiten grabar el actual estado de la partida.

Si desea terminar la partida en ese momento, entre simplemente SAVE. Los programas más elaborados le preguntarán entonces por un nombre, bajo el cual se memorizará la actual situación, mientras que los programas sencillos procesarán esta instrucción inmediatamente.

También es aconsejable grabar la partida de vez en cuando. Podría presentarse, por ejemplo, la situación en que usted pretendiese escalar una peña cortada a pique. Es muy probable que usted caiga y se desnuque; la consecuencia de ello sería volver a empezar de nuevo y repefir todas las entradas.

¡Una grabación a tiempo le hubiera evitado esta molestia poco interesante!

REANUDAR UNA PARTIDA INTERRUMPIDA

Debe introducir por supuesto la correspondiente cinta o diskette antes de entrar simplemente LOAD. Es posible que le pregunten por el nombre con el que grabó el estado de la partida. De lo contrario, bastará pulsar la tecla <RETURN> para reanudar el juego en el lugar interrumpido.

INDICACIONES ADICIONALES

En primer lugar, usted deberá averiguar a qué tipo de aventura pertenece la que acaba de cargar en su ordenador, si su imagen titular no le ha informado sobre ello.

Tendrá que cumplir alguna misión, encontrar alguna salida o coleccionar tesoros.

Inspeccione todo lo que vea y actúe siempre de forma lógica (para ABRIR una puerta CERRADA, necesitará primero una LLAVE, una BARRA de hierro o algo parecido) al menos hasta que sea informado de si hay magia en juego. Entonces también podrá efectuar entradas absurdas, y estudiar la solución correcta lo antes posible.

Preste atención sobre todo, a informaciones del tipo ESTO NO ES POSIBLE DE MOMENTO que le indican que el camino tomado es correcto.

Pero desgraciadamente, no se han cumplido aún todas las condiciones necesarias para realizar la acción (por ejemplo, faltará la llave).

Si el verbo o el objeto no ha sido entendido, se le informará de ello y usted deberá intentar describir la misma acción con distintas palabras.

Puede darse una última posibilidad que se expresa mediante NO ENTIENDO LO QUE QUIERES DECIR. Esto significa, que las palabras que usted haya utilizado en su entrada, son perfectamente conocidas, mientras que su combinación carece de sentido. También puede significar que la acción descrita no ha sido prevista por el programador de la aventura en cuestión.

CONSEJOS PARA SOLUCIONAR UNA AVENTURA

Si ya no sabe qué hacer en situaciones muy enmarañadas,

puede probar su suerte con HELP.

Si esto no le proporcionará más información, recuerde todos los objetos descubiertos hasta ese momento y combínelos con todos los verbos conocidos. Quizás de esta forma pueda encontrar una instrucción que le permita realizar la acción deseada, puesto que a cada objeto le corresponde realizar alguna misión. No están programados con otra intención.

Resulta útil confeccionar una lista de todas las palabras utilizadas en el programa; probablemente, la misma aventura se la proporcione al efectuar la entrada VOCabulario.

También un mapa puede ser de utilidad. En el puede indicar todos los lugares, sin olvidar naturalmente, los sitios donde ha hallado los tesoros.

Si se para de manera inesperada - o también deseada, por ser inevitable - en un laberinto, la confección de este tipo de mapa no resulta tan fácil como pueda imaginarse. En este caso, suelen catalogarse tres veces más lugares de los realmente existentes.

El único truco para evitar esto, consiste en depositar un objeto del inventario en cada lugar del laberinto, señalándolos de esta forma.

Repetimos otra vez el truco más importante:
¡Grabe su partida de vez en cuando!

EL CASTILLO ENCANTADO

Quiero recomendarle especialmente este juego de aventuras. En su confección se han utilizado muchos trucos que permiten ofrecer sorpresas al aventurero.

En principio, está previsto que el jugador averigue el objetivo de la aventura, problema que usted ya resolverá al entrar el programa en el ordenador.

En este programa se ha utilizado gran número de las técnicas presentadas en este libro. A continuación mencionamos algunas secciones a modo de ejemplo.

De entrada destaca la línea 181, donde puede ver la forma en que deberá proceder si las tres líneas de pantalla, programadas para efectuar las entradas de las descripciones neceasarias, no son suficientes.

Además destacan las líneas 600 a 610. En ellas se emiten mensajes llamando los correspondientes subprogramas, técnica que se debería emplear con mensajes especialmente repetitivos.

En este caso se utilizaron flags para poder controlar perfectamente los diferentes estados antes de abandonar el castillo.

Cuando se cumplan todas las condiciones, se abrirá un paso adecuado, que ampliará el mundo de aventuras a 16 lugares.

Además de INVENTARIO, SAVE y LOAD, también se ha implementado la instrucción HELP, aunque sólo de forma general.

No obstante, el jugador sabrá apreciar esta ayuda.

```
1 REM
2 REM
4 REM
10 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:FOR I=1 TO
5:PRINT :NEXT I
15 PRINT "                                    "
16 PRINT "                                     "
17 PRINT "                                     "
18 PRINT "                                     "
19 PRINT "                                     "
20 PRINT "                                     "
21 PRINT "          CASTILLO                   "
22 PRINT "          ENCANTADO                  "
23 PRINT "                                      "
24 PRINT "          UNA AVENTURA               "
25 PRINT "               DE                    "
26 PRINT "          FERRE MORET"
27 PRINT "                                    ";
100 REM *** DATEN DES ADVENTURES
110 AR=26:AO=45:AF=3
120 JUGADOR=9
140 TRAP PEEK(195)
150 DIM
LUGAR$(80),PASADA(AR,6),DIRECCION$(36),ENTRADA$(20),T$(40),OB
(AO),BLANCO$(38),DZ$(50),ZENDE$(3),FL(5)
160 DIM
VERBOS$(80),EVERBO$(20),EOBJETO$(20),OBJETOS$(220),MO$(40),M1
$(30),M2$(30),M3$(30),M4$(38),M5$(38)
```

```
170 VERBOS$="EXAMINA   COGE      LEE       ABRE      UTILIZA
DESTRUYE  ENCIENDE  PON       DESPIERTA "
180 OBJETOS$="ESTAMESAARI
CACEBOTEFUENLIBRGUARCUERHABIHUESPAPECOCICUCHBALLLLAVTAL PUEN
ARBO    ESTA    ARBO"
181 OBJETOS$(LEN(OBJETOS$)+1)="         URRACAVE OROARMA
PANTBARRHUEVSUELAGUAPOZO    TELABOTE    NIDOSERPALTA"
190 GOTO 700
200 REM ***** DESCRIPCION DE LUGARES
201 PRINT "EN LA BIBLIOTECA DEL CASTILLO":RETURN
202 PRINT "EN EL CUARTO DE ESTUDIOS.":RETURN
203 PRINT "EN LA COCINA.":RETURN
204 PRINT "EN EL PATIO DEL CASTILLO.":RETURN
205 PRINT "EN UNA DE LAS TORRES DE GUARDIA.":RETURN
206 PRINT "EN EL PATIO DEL CASTILLO.":RETURN
207 PRINT "DELANTE DEL PUENTE ELEVADIZO.":RETURN
208 PRINT "EN LA GRAN SALA DE FIESTAS.":RETURN
209 PRINT "EN UNA HABITACION PARTICULAR.":RETURN
210 PRINT "SOBRE UN PUENTE ELEVADIZO.":RETURN
211 PRINT "EN EL PATIO DEL CASTILLO.":RETURN
212 PRINT "EN UN CAMINO DEL BOSQUE.":RETURN
213 PRINT "EN EL BOSQUE ENCANTADO.":RETURN
214 PRINT "EN UNA OSCURA CAVERNA.":RETURN
215 PRINT "EN EL BOSQUE.":RETURN
216 PRINT "EN EL BOSQUE ENCANTADO.":RETURN
217 PRINT "EN EL BOSQUE ENCANTADO.":RETURN
218 PRINT "AL PIE DE UNA CORDILLERA.":RETURN
219 PRINT "EN UNA OSCURA CAVERNA.":RETURN
```



```
220 PRINT "EN UN CASTILLO.":RETURN
221 PRINT "EN EL BOSQUE.":RETURN
222 PRINT "EN EL PANTANO.":RETURN
223 PRINT "EN EL PANTANO.":RETURN
224 PRINT "EN UNA FUENTE.":RETURN
225 PRINT "EN UNA OSCURA CAVERNA.":RETURN
226 PRINT "EN EL POZO DE LAS SERPIENTES.":RETURN
300 REM ***** OBJETOS
301 T$="INNUMERABLES ESTANTERIAS":RETURN
302 T$="UNA MESA":RETURN
303 T$="MI PROFESOR ARI AT":RETURN
304 T$="UNA GRAN CACEROLA":RETURN
305 T$="UNA BOTELLA":RETURN
306 T$="LA FUENTE DEL CASTILLO":RETURN
307 T$="UN LIBRO":RETURN
308 T$="GUARDIAS DURMIENDO":RETURN
309 T$="LAS CUERDAS DEL PUENTE ELEVADIZO":RETURN
310 T$="HABITANTES DEL CASTILLO":RETURN
311 T$="MIS HUESPEDES":RETURN
312 T$="UN PAPEL":RETURN
313 T$="UNA MESA DE LA COCINA":RETURN
314 T$="UN CUCHILLO DE CARNICERO":RETURN
315 T$="UNA BALLESTA":RETURN
316 T$="UNA LLAVE":RETURN
317 T$="UN PESADO PORTAL DE HIERRO":RETURN
318 T$="UN PUENTE ELEVADIZO":RETURN
319 T$="NADA ESPECIAL":RETURN
320 T$="NADA ESPECIAL":RETURN
321 T$="ARBOLES":RETURN
322 T$="ARBOLES":RETURN
323 T$="VARIAS ESTANTERIAS LLENAS":RETURN
324 T$="NADA ESPECIAL":RETURN
325 T$="ARBOLES":RETURN
326 T$="ARBOLES":RETURN
```

```
327 T$="ARBOLES":RETURN
328 T$="UNA GRAN URRACA":RETURN
329 T$="LA ENTRADA DE UNA CAVERNA":RETURN
330 T$="UN METALL CON RESPLANDOR DE ORO":RETURN
331 T$="UNA ARMADURA VIEJA":RETURN
332 T$="NADA ESPECIAL":RETURN
333 T$="UN PANTANO":RETURN
334 T$="BARRO":RETURN
335 T$="HUEVOS DE SERPIENTE":RETURN
336 T$="EL SUELO INSEGURO":RETURN
337 T$="AGUA TORRENCIAL":RETURN
338 T$="EL POZO DE UN CASTILLO":RETURN
339 T$="NADA ESPECIAL":RETURN
340 T$="TELARANYAS":RETURN
341 T$="UNA BOTELLA CON AGUA AZUL":RETURN
342 T$="NADA ESPECECIAL":RETURN
343 T$="UN NIDO DE SERPIENTES":RETURN
344 T$="SERPIENTES":RETURN
345 T$="UN ALTAR DE PIEDRA":RETURN
400 REM ***** POSICION OBJETOS
410 DATA
1,1,2,3,0,4,0,5,5,8,8,0,3,0,5,0,6,7,9,11,12,13,14,25,15,16,17
,17,18,19,0,21,0,23,26,23,24,10,20,0,0,19,22
420 DATA 26,25
500 REM ***** TABLA DE DIRECCIONES
501 DATA 0,3,0,2,0,0
502 DATA 0,0,1,0,0,0
503 DATA 1,8,0,4,0,0
504 DATA 0,8,3,6,0,0
505 DATA 0,0,0,0,0,11
506 DATA 11,0,4,0,0,0
507 DATA 0,0,6,0,0,0
508 DATA 4,8,3,9,0,0
509 DATA 9,9,8,0,0,0
510 DATA 0,0,7,12,0,0
511 DATA 0,6,0,0,5,0
512 DATA 12,16,10,13,0,0
```



```
513 DATA 21,17,12,15,0,0
514 DATA 0,18,14,0,0,0
515 DATA 15,15,15,16,0,0
516 DATA 12,21,15,17,0,0
517 DATA 13,22,16,18,0,0
518 DATA 14,0,17,19,0,0
519 DATA 25,0,18,19,0,0
520 DATA 20,20,20,21,0,0
521 DATA 16,21,20,21,0,0
522 DATA 17,0,0,23,0,0
523 DATA 0,0,22,24,0,0
524 DATA 23,24,0,23,0,0
525 DATA 25,19,25,14,14,0
526 DATA 0,0,0,0,22,0
600 PRINT "NO ENTIENDO LO QUE QUIERES!":RETURN
601 PRINT "NO VEO NADA ESPECIAL.":RETURN
602 PRINT "NO SOY TAN FUERTE.":RETURN
603 PRINT "PARECEN DORMIR PROFUNDAMENTE.":RETURN
604 PRINT "NO CREO QUE SEA MUY RAZONABLE.":RETURN
606 PRINT "LA URRACA ME HA ROBADO ALGO !":RETURN
607 PRINT "AQUI HAY UNA VIEJA ARMADURA.":RETURN
608 PRINT "TIENE UN PAPEL EN LA MANO.":RETURN
609 PRINT "EN UN BOLSILLO HAY UNA LLAVE.":RETURN
610 PRINT "HAY UNA INSCRIPCION QUE DICE:":PRINT "ACEPTA
NUESTRO SACRIFICIO Y TEN PIEDAD DE NOSOTROS.":RETURN
699 REM ***** TITEL:EINLEITUNG
700 PRINT " ":PRINT :PRINT " DESEA CONSEJOS PARA
SEGUIR";:INPUT ENTRADA$
798 IF ENTRADA$(1,1)="S" THEN GOSUB 800
```

```
799 GOTO 900
800 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0
820 PRINT "              ATARI AVENTURAS          ":PRINT "
(C) 1984 BY JOERG WALKOWIAK    ":REM INVERS
830 PRINT :PRINT "Imaginese un robot al cual puede
dirigir mediante numerosos comandos."
840 PRINT "Yo soy tal robot y correre para Ud."
850 PRINT "todos los peligros en las aventuras   mas
arriesgadas."
855 PRINT "Para que Ud. pueda manejarme de forma razonable,le
describire con detalle"
860 PRINT "la situacion en la que me encuentre.":PRINT "Acto
seguido Ud. me ordenara mediante"
863 PRINT "dos palabras, p. e."
870 PRINT "EXAMINA PUERTA, COGE CUCHILLO, lo que yo deba
hacer.":PRINT "Ademas entiendo las ordenes"
880 PRINT "   INVENTARIO y AYUDA"
885 PRINT "          SAVE, LOAD,"
887 PRINT "        asi como FIN.":PRINT
890 PRINT "POR FAVOR TECLEE <RETURN> Y...";:INPUT
ENTRADA$:GRAPHICS 0:RETURN :REM INVERSO
900 FOR I=1 TO AO
910 READ PARADERO:OB(I)=PARADERO
920 NEXT I
930 FOR I=1 TO AF:FL(I)=0:NEXT I
940 FOR LUGAR=1 TO AR
950 FOR DIRECCION=1 TO 6
960 READ META:PASADA(LUGAR,DIRECCION)=META
```



```
970 NEXT DIRECCION
980 NEXT LUGAR
1000 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:GOTO 1020
1010 REM ***** TEXTO CLARO:DIRECCIONES
1011 T$="NORTE, ":RETURN
1012 T$="SUR, ":RETURN
1013 T$="OESTE, ":RETURN
1014 T$="ESTE, ":RETURN
1015 T$="ARRIBA, ":RETURN
1016 T$="ABAJO, ":RETURN
1020 BLANCO$="                                        "
1030 T$=""
1040
ZENDE$=CHR$(30):ZENDE$(LEN(ZENDE$)+1)=CHR$(30):ZENDE$(LEN(ZEN
DE$)+1)="."
1080 PRINT
1090 FOR LINEA=0 TO 10:POSITION 2,LINEA:PRINT BLANCO$:NEXT
LINEA
1100 POSITION 2,0:PRINT "ESTOY EN ";:GOSUB 200+JUGADOR
1110 DZ$="VEO "
1120 FOR I=1 TO AO
1130 IF OB(I)<>JUGADOR THEN GOTO 1170
1140 IMPRESO=-1:GOSUB 300+I:T$(LEN(T$)+1)=", "
1150 IF LEN(DZ$)+LEN(T$)>=38 THEN PRINT
DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 1170
1160 DZ$(LEN(DZ$)+1)=T$:T$=""
1170 NEXT I
1180 IF  NOT IMPRESO THEN DZ$(LEN(DZ$)+1)="NADA ESPECIAL"
1190 DZ$(LEN(DZ$)+1)=ZENDE$
1200 PRINT DZ$
1210 PRINT BLANCO$
1220 DZ$="PUEDO IR AL ":IMPRESO=0
1230 FOR DIRECCION=1 TO 6
1240 IF PASADA(JUGADOR,DIRECCION)=0 THEN GOTO 1280
```

```
1250 GOSUB 1010+DIRECCION:IMPRESO=-1
1260 IF LEN(DZ$)+LEN(T$)>=38 THEN PRINT
DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 1280
1270 DZ$(LEN(DZ$)+1)=T$:T$=""
1280 NEXT DIRECCION
1290 IF  NOT IMPRESO THEN DZ$(LEN(DZ$)+1)="EN NINGUNA PARTE"
1300 DZ$(LEN(DZ$)+1)=ZENDE$
1310 PRINT DZ$
1320 PRINT "                                    ":REM 37 CONTROL R
1340 IF JUGADOR=25 THEN GOSUB 14000
1390 POSITION 2,23:PRINT "QUE HAGO ";:INPUT ENTRADA$
1392 IF JUGADOR=17 THEN GOSUB 15000
1394 IF JUGADOR=23 THEN GOSUB 15100
1396 IF JUGADOR=15 THEN GOSUB 15200
1400 IF LEN(ENTRADA$)>2 THEN 1480
1410 IF ENTRADA$="N" AND PASADA(JUGADOR,1)<>0 THEN
JUGADOR=PASADA(JUGADOR,1):PRINT "O.K.":GOTO 1080
1420 IF ENTRADA$="S" AND PASADA(JUGADOR,2)<>0 THEN
JUGADOR=PASADA(JUGADOR,2):PRINT "O.K.":GOTO 1080
1430 IF ENTRADA$="O" AND PASADA(JUGADOR,3)<>0 THEN
JUGADOR=PASADA(JUGADOR,3):PRINT "O.K.":GOTO 1080
1440 IF ENTRADA$="E" AND PASADA(JUGADOR,4)<>0 THEN
JUGADOR=PASADA(JUGADOR,4):PRINT "O.K.":GOTO 1080
1450 IF ENTRADA$="AR" AND PASADA(JUGADOR,5)<>0 THEN
JUGADOR=PASADA(JUGADOR,5):PRINT "O.K.":GOTO 1080
1460 IF ENTRADA$="AB" AND PASADA(JUGADOR,6)<>0 THEN
JUGADOR=PASADA(JUGADOR,6):PRINT "O.K.":GOTO 1080
1470 IF LEN(ENTRADA$)<3 THEN PRINT "HACIA ALLI NO HAY NINGUN
CAMINO !":GOTO 1080
1480 IF LEN(ENTRADA$)>8 THEN GOTO 2000
1499 REM ***** COMIENZO INVENTARIO
```



```
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO 1600
1510 PRINT "LLEVO CONMIGO LO SIGUIENTE:"
1520 FOR I=1 TO AO
1530 IF OB(I)=-1 THEN GOSUB 300+I:PRINT T$
1540 NEXT I
1550 GOTO 1080
1560 REM ***** FIN INVENTARIO
1599 REM ***** SAVE GAME
1600 IF ENTRADA$(1,3)<>"SAV" THEN GOTO 1700
1605 PRINT "CON QUE NOMBRE";:INPUT ENTRADA$:IF
LEN(ENTRADA$)>8 THEN PRINT "MAS CORTO POR FAVOR !":GOTO 1605
1610 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)=".DAT"
1615 T$(LEN(T$)+1)=ENTRADA$
1620 OPEN#1,8,0,T$
1625 PRINT#1,JUGADOR
1630 FOR I=1 TO AO
1631 PRINT#1,OB(I)
1632 NEXT I
1635 FOR LUGAR=1 TO AR
1636 FOR DIRECCION=1 TO 6
1637 PRINT    #1,PASADA(LUGAR,DIRECCION)        1638 NEXT
DIRECCION
1639 NEXT LUGAR
1645 FOR I=1 TO AF
1646 PRINT#1,FL(I)
1647 NEXT I
1650 CLOSE#1
1670 PRINT "O.K.":GOTO 1080
1699 REM ***** LOAD GAME
1700 IF ENTRADA$(1,3)<>"LOA" THEN GOTO 1800
1705 PRINT "QUE JUEGO ";:INPUT ENTRADA$:IF LEN(ENTRADA$)>8
THEN PRINT "NO PUEDE HABERLO !":GOTO 1705
1710 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)=".DAT"
```

```
1715 T$(LEN(T$)+1)=ENTRADA$
1720 OPEN# 1,4,0,T$
1725 INPUT# 1,JUGADOR
1730 FOR I=1 TO AO
1731 INPUT # 1,SITUACION:OB(I)=SITUACION
1732 NEXT I
1735 FOR LUGAR=1 TO AR
1736 FOR DIRECCION=1 TO 6
1737 INPUT#1,META:PASADA(LUGAR,DIRECCION)=META
1738 NEXT DIRECCION
1739 NEXT LUGAR
1745 FOR I=1 TO AF
1746 INPUT# 1,FLAG:FL(I)=FLAG
1747 NEXT I
1750 CLOSE #1
1770 PRINT "O.K.":GOTO 1080
1800 IF ENTRADA$(1,3)<>"AYU" THEN GOTO 1900
1810 GOSUB 1880
1820 GEDRUCKT=0
1830 FOR I=1 TO LEN(VERBOS$) STEP 10
1840 PRINT VERBOS$(I,I+9):IMPRESO=IMPRESO+1
1850 IF IMPRESO=18 THEN GOSUB 1890:IMPRESO=1:GOSUB 1880
1860 NEXT I
1870 GOSUB 1890:GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:GOTO
1080
1880 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:PRINT "INTENTA
LAS SIGUIENTES ACCIONES...":PRINT :RETURN
1890 PRINT :PRINT :PRINT "APRETAR <RETURN>...";:INPUT
ENTRADA$:RETURN
1900 IF ENTRADA$(1,3)<>"INS" THEN GOTO 1950
1910 GOSUB 800
1920 GOTO 1080
1950 IF ENTRADA$(1,3)<>"FIN" THEN GOTO 2000
```



```
1960 GRAPHICS 0:PRINT "EL AUTOR LE DESEA MAS SUERTE PARA
LA":PRINT "PROXIMA VEZ!":PRINT :PRINT :PRINT :END
1999 REM ***** SEPARAR ENTRADA
2000 LN=LEN(ENTRADA$)
2010 FOR I=1 TO LN
2020 IF ENTRADA$(I,I)<>" " THEN NEXT I
2030 EVERBO$=ENTRADA$(1,I)
2040 IF LEN(EVERBO$)=LN-1 THEN GOTO 2090
2050 EOBJETO$=ENTRADA$(I+1,LN)
2060 VN=0:N=0
2070 EVERBO$=EVERBO$(1,3)
2080 EOBJETO$=EOBJETO$(1,3)
2090 FOR I=1 TO LEN(VERBOS$)-2 STEP 10
2100 VN=VN+1
2110 IF VERBOS$(I,I+2)=EVERBO$ THEN 2140
2120 NEXT I
2130 PRINT "NO ENTIENDO EL VERBO !":GOTO 1080
2140 FOR I=1 TO LEN(OBJETOS$) STEP 4
2150 N=N+1
2160 IF OBJETOS$(I,I+2)=EOBJETO$ THEN GOTO 2200
2170 NEXT I
2180 PRINT "NO COMPRENDO EL OBJETO.":GOTO 1080
2200 ON VN GOTO
5000,6000,7000,8000,9000,10000,11000,12000,16000
4500 GRAPHICS 0:REM ***** MUERTE JUGADOR
4600 PRINT "LO QUE FALTABA!":PRINT :PRINT MO$
4610 PRINT :PRINT "ESTOY MUERTO!":PRINT
4620 PRINT "QUIERE QUE LO INTENTE DE NUEVO ";:INPUT ENTRADA$
4630 IF ENTRADA$(1,1)="S" THEN CLR :GOTO 100
4640 GOTO 1960
```

```
4800 GRAPHICS 0
4810 PRINT "FELICIDADES !"
4820 PRINT "HA ENCONTRADO LA SOLUCION DEL PROBLEMA":PRINT "Y
PUEDE INTENTAR UNA NUEVA AVENTURA."
4840 PRINT :PRINT :PRINT :END
4900 PRINT "ATENCION ERROR !":GOTO 1080
4999 REM ***** EJECUCION DE LAS JUGADAS
5000 IF OB(N)<>JUGADOR AND OB(N)<>-1 THEN GOTO 5900
5001 IF N=16 AND JUGADOR=20 AND OB(31)<>-1 AND OB(40)<>-1
THEN GOSUB 607:OB(31)=20:OB(40)=20:GOTO 1080
5002 IF N=10 AND JUGADOR=8 THEN GOSUB 603:GOTO 1080
5003 IF N=11 AND JUGADOR=8 THEN GOSUB 603:GOTO 1080
5004 IF N=4 AND (JUGADOR=3 OR OB(4)=-1) THEN GOSUB 601:GOTO
1080
5005 IF N=5 AND (JUGADOR=3 OR OB(N)=-1) THEN GOSUB 601:GOTO
1080
5006 IF N=1 AND JUGADOR=1 THEN GOSUB 601:GOTO 1080
5008 IF N=3 AND JUGADOR=2 AND OB(12)=0 THEN GOSUB
608:OB(12)=2:GOTO 1080
5009 IF N=3 AND JUGADOR=2 THEN GOSUB 601:GOTO 1080
5010 IF N=6 AND OB(5)=0 THEN PRINT "SOBRE EL AGUA FLOTA UNA
BOTELLA.":OB(5)=JUGADOR:GOTO 1080
5011 IF N=7 AND (JUGADOR=1 OR OB(N)=-1) THEN PRINT "ES UN
LIBRO SOBRE POCIONES MAGICAS.":GOTO 1080
5012 IF N=8 AND JUGADOR=5 AND OB(16)=0 THEN GOSUB
609:OB(16)=JUGADOR:GOTO 1080
5013 IF N=9 AND  NOT FL(1) THEN GOSUB 601:GOTO 1080
```



```
5014 IF N=9 AND FL(1) AND JUGADOR=5 THEN PRINT "NO ESTA EN
CONDICIONES DE FUNCIONAR.":GOTO 1080
5015 IF N=12 AND OB(12)=-1 THEN PRINT "HAY ALGO ESCRITO
ENCIMA.":OB(16)=JUGADOR:GOTO 1080
5016 IF N=12 AND OB(12)=2 THEN PRINT "ENTONCES PRIMERO TENGO
QUE COGERLO.":GOTO 1080
5017 IF N=13 AND OB(14)=0 THEN PRINT "SOBRE LA MESA HAY UN
CUCHILLO.":OB(14)=3:GOTO 1080
5018 IF N=2 AND (OB(7)=1 OR OB(7)=0) THEN PRINT "HAY UN LIBRO
VIEJO ENCIMA.":OB(7)=1:GOTO 1080
5019 IF N=14 AND (OB(14)=JUGADOR OR OB(14)=-1) THEN PRINT
"ESTA MUY AFILADO.":GOTO 1080
5020 IF N=15 AND (OB(N)=JUGADOR OR OB(N)=-1) THEN GOSUB
601:GOTO 1080
5021 IF N=16 AND (OB(16)=-1 OR OB(16)=JUGADOR) THEN GOSUB
601:GOTO 1080
5022 IF N=17 AND JUGADOR=6 AND FL(3)<>-1 THEN PRINT "ESTA
CERRADA.":GOTO 1080
5023 IF N=17 AND JUGADOR=6 AND FL(3)=-1 THEN PRINT "ESTA MUY
ABIERTA.":GOTO 1080
5024 IF N=18 AND JUGADOR=7 AND FL(1) THEN PRINT "EL PUENTE
ELEVADIZO BAJA.":PASADA(7,4)=10:GOTO 1080
5025 IF N=18 AND JUGADOR=7 AND  NOT FL(1) THEN PRINT "EL
PUENTE ELEVADIZO ESTA SUBIDO.":GOTO 1080
5026 IF N=21 THEN GOSUB 601:GOTO 1080
5027 IF N=23 AND JUGADOR=14 THEN MO$="ES UN LABORATRIO
MAGICO. EL MAGO VIENE Y ME TRANSFORMA EN PIEDRA.":GOTO 4500
5028 IF N=28 AND JUGADOR=17 THEN PRINT "REALMENTE ES UN
EJEMPLAR HERMOSO.":GOSUB 13000:GOTO 1080
```

```
5029 IF N=29 AND JUGADOR=18 THEN PRINT "SALE UN OLOR
RARO.":GOTO 1080
5030 IF N=30 AND (JUGADOR=19 OR OB(30)=-1) THEN PRINT "EL
METAL SIN NINGUN VALOR.":GOTO 1080
5031 IF N=31 AND (JUGADOR=20 OR OB(31)=-1) THEN GOSUB
601:GOTO 1080
5032 IF N=35 AND (JUGADOR=26 OR OB(35)=-1) THEN GOSUB
601:GOTO 1080
5033 IF N=36 AND JUGADOR=23 THEN PRINT "NO PARECE MUY
FIRME.":GOTO 1080
5034 IF N=37 AND JUGADOR=24 THEN PRINT "ES EL FAMOSO AGUA
AZUL.":GOTO 1080
5035 IF N=38 AND JUGADOR=10 THEN GOSUB 601:GOTO 1080
5036 IF N=40 AND (JUGADOR=20 OR OB(40)=-1) THEN GOSUB
601:GOTO 1080
5037 IF N=43 THEN JUGADOR=26:PRINT "O.K.":GOTO 1080
5038 IF N=45 THEN GOSUB 610:GOTO 1080
5899 GOSUB 601:GOTO 1080
5900 IF N=21 THEN GOSUB 601:GOTO 1080
5901 IF N=16 AND JUGADOR=20 AND OB(31)<>-1 AND OB(40)<>-1
THEN GOSUB 607:OB(31)=20:OB(40)=20:GOTO 1080
5902 IF N=33 AND JUGADOR=23 THEN GOSUB 15100:PRINT "PARECE
QUE HAY UNA FUENTE AL ESTE.":GOTO 1080
5903 IF N=16 AND (JUGADOR=23 OR OB(34)=-1) THEN PRINT "ES
BARRO TERAPEUTICO.":GOSUB 15100:GOTO 1080
5904 IF N=2 AND JUGADOR=3 AND OB(14)=0 THEN PRINT "SOBRE LA
MESA HAY UN CUCHILLO.":OB(14)=3:GOTO 1080
5905 IF N=2 AND JUGADOR=3 AND OB(14)<>0 THEN GOSUB 601:GOTO
1080
5990 PRINT "AQUI NO VEO TAL COSA !":GOTO 1080
```



```
6000 IF OB(N)<>JUGADOR AND OB(N)<>-1 THEN GOTO 6900
6001 IF N=1 THEN GOSUB 602:GOTO 1080
6002 IF N=6 THEN GOSUB 602:GOTO 1080
6003 IF N=17 THEN GOSUB 602:GOTO 1080
6004 IF N=2 THEN GOSUB 604:GOTO 1080
6005 IF N=3 THEN GOSUB 604:GOTO 1080
6006 IF N=8 THEN GOSUB 604:GOTO 1080
6007 IF N=10 THEN GOSUB 604:GOTO 1080
6008 IF N=11 THEN GOSUB 604:GOTO 1080
6009 ZA=0:FOR I=1 TO AO:IF OB(I)=-1 THEN ZA=ZA+1:IF ZA=4 THEN
GOTO 6999
6010 NEXT I
6011 IF N=5 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6012 IF N=7 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6013 IF N=12 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6014 IF N=15 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6015 IF N=16 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6016 IF N=14 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6017 IF N=9 THEN GOSUB 602:GOTO 1080
6018 IF N=23 THEN GOSUB 602:GOTO 1080
6019 IF N=28 THEN PRINT "CON QUE LA CAZO ?":GOSUB 13000:GOTO
1080
6020 IF N=18 THEN GOSUB 600:GOTO 1080
6021 IF N=30 THEN OB(N)=-1:PRINT "O.K.":GOTO 1080
6022 IF N=31 THEN OB(31)=-1:PRINT "ME HE PUESTO LA
ARMADURA.":GOTO 1080
6023 IF N=35 AND OB(31)=-1 THEN PRINT "O.K.":OB(35)=-1:GOTO
1080
6024 IF N=35 AND OB(31)<>-1 THEN MO$="ME HA MORDIDO UNA
SERPIENTE.":GOTO 4500
```

```
6025 IF N=36 THEN GOSUB 604:GOTO 1080
6026 IF N=37 AND OB(5)=-1 THEN PRINT
"O.K.":OB(5)=0:OB(41)=-1:GOTO 1080
6027 IF N=37 AND OB(5)<>-1 THEN PRINT "ANTES NECESITO UNA
BOTELLA VACIA.":GOTO 1080
6028 IF N=38 THEN GOSUB 600:GOTO 1080
6029 IF N=40 THEN OB(40)=-1:PRINT "O.K.":GOTO 1080
6032 IF N=13 THEN GOSUB 604:GOTO 1080
6033 IF N=4 THEN OB(4)=-1:PRINT "O.K.":GOTO 1080
6900 IF N=16 AND JUGADOR=23 AND OB(4)<>-1 THEN PRINT "ANTES
NECESITO UN RECIPIENTE !":GOTO 1080
6901 IF N=16 AND JUGADOR=23 AND OB(4)=-1 THEN PRINT
"O.K.":OB(34)=-1:GOTO 1080
6902 IF N=2 AND JUGADOR=3 THEN GOSUB 604:GOTO 1080
6998 PRINT "AQUI NO VEO TAL COSA !":GOTO 1080
6999 PRINT "LO SIENTO-PERO YA LLEVO BASTANTE ":GOTO 1080
7000 IF N=13 AND OB(12)<>-1 THEN PRINT "NO TENGO NINGUN
PAPEL.":GOTO 1080
7001 IF N=12 AND OB(12)=-1 THEN CO=INT(RND(1)*100):PRINT "VEO
UNA ESCRITURA TEMBLOROSA:";CO:GOTO 1080
7002 IF N=7 AND OB(7)=-1 THEN GOTO 7650
7003 IF N=7 AND OB(7)<>-1 THEN PRINT "PARA ELLO PRIMERO TENGO
QUE TENERLO.":GOTO 1080
7649 GOSUB 600:GOTO 1080
7650 PRINT "QUE LADO";:INPUT SE
7651 IF SE<>CO THEN GOSUB 601:GOTO 1080
7652 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:PRINT "CONSEJOS
PARA EMERGENCIAS":PRINT
```



```
7654 PRINT "SI OCURRIERE QUE LOS TUYOS FUESEN":PRINT
"SOMETIDOS POR UN MAGO DE MAL TALANTE"
7655 PRINT "AL LARGO SUENYO, ESCUCHA MI CONSEJO":PRINT
7657 PRINT :PRINT "ENCUENTRA LOS CUATRO INGREDIENTES":PRINT
"PARA EL AGUA QUE CURA DE HMBUG Y"
7659 PRINT "LLEVASELOS AL MAGO DEL OTRO LADO DEL":PRINT
"BOSQUE MAGICO."
7660 PRINT "SI LE CAES BIEN, EL TE AYUDARA."
7661 PRINT "SI NO LE CAES BIEN SOLO TE QUEDA LA":PRINT
"ESPERANZA DE ENCONTRARTE CON UN PRINCIPE,"
7663 PRINT "EL CUAL A MODO TRADICIONAL DESPERTARA":PRINT "A
TU HIJA Y A TODOS LOS"
7665 PRINT "HABITANTES DEL CASTILLO !"
7666 OPEN#1,4,0,"K:"
7667 GET#1,A
7668 CLOSE #1
7669 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:GOTO 1080
7900 OB(N)=JUGADOR:PRINT "O.K.":GOTO 1080
8000 IF N=17 AND OB(16)<>-1 THEN PRINT "CON QUE ?":GOTO 1080
8001 IF N=17 AND OB(16)=-1 THEN PRINT
"O.K.":PASADA(6,4)=7:FL(3)=-1:GOTO 1080
8002 IF N=35 AND OB(35)=-1 THEN PRINT "DENTRO HABIA UN
LIQUIDO VISCOSO.":GOTO 1080
8999 PRINT "NO ENTIENDO LO QUE QUIERES.":GOTO 1080
9000 IF N=9 AND OB(14)=-1 AND JUGADOR=5 THEN PRINT "LAS
CUERDAS ESTAN CORTADAS.":FL(1)=-1:GOTO 1080
```

```
9001 IF N=9 AND OB(14)<>-1 AND JUGADOR=5 THEN PRINT "CON
QUE?":GOTO 1080
9999 GOSUB 600:GOTO 1080
10000 IF OB(N)<>JUGADOR AND OB(N)<>-1 THEN PRINT "AQUI NO VEO
TAL COSA.":GOTO 1080
10001 IF N=9 THEN PRINT "PARA ESTO HACEN FALTA COMO
MINIMO":PRINT "DOS HOMBRES.":GOTO 1080
10002 IF N=14 AND JUGADOR=5 THEN PRINT "LAS CUERDAS ESTAN
CORTADAS.":FL(1)=-1:GOTO 1080
10003 IF N=31 THEN PRINT "O.K.- LLEVO LA
ARMADURA.":OB(31)=-1:GOTO 1080
10990 GOSUB 600:GOTO 1080
11000 IF N=5 AND JUGADOR=24 AND OB(5)=-1 THEN PRINT
"O.K.":OB(5)=0:OB(41)=-1:GOTO 1080
11001 IF N=5 AND JUGADOR<>24 AND OB(5)=-1 THEN PRINT "CON
QUE?":GOTO 1080
11990 GOSUB 600:GOTO 1080
12000 IF OB(N)<>-1 THEN GOTO 12900
12001 IF N=4 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12002 IF N=5 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12003 IF N=7 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12004 IF N=12 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12005 IF N=15 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12006 IF N=16 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12007 IF N=14 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12008 IF N=30 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12009 IF N=41 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
```



```
12010 IF N=45 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12011 IF N=40 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12012 IF N=36 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12013 IF N=35 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12014 IF N=31 THEN PRINT "O.K.":OB(N)=JUGADOR:GOTO 1080
12900 IF N=5 AND OB(41)=-1 THEN PRINT
"O,K.":OB(41)=JUGADOR:GOTO 1080
12901 IF N=16 AND OB(34)=-1 THEN PRINT
"O.K.":OB(34)=JUGADOR:GOTO 1080
12990 GOSUB 600:GOTO 1080
13000 IF OB(30)=-1 THEN OB(30)=20:GOSUB 606:RETURN
13010 IF OB(35)=-1 THEN OB(35)=20:GOSUB 606:RETURN
13020 IF OB(5)=-1 THEN OB(5)=20:GOSUB 606:RETURN
13090 RETURN
14000 IF OB(34)<>25 THEN RETURN
14010 IF OB(35)<>25 THEN RETURN
14020 IF OB(41)<>25 THEN RETURN
14030 IF OB(40)<>25 THEN RETURN
14100 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR 4,1,0:PRINT "SE OYE
UNA VOZ:":PRINT :PRINT :PRINT
14110 PRINT "EL GRAN HUMB SE ALEGRA DE TUS DONES.":PRINT
14120 PRINT "HARA SERVIR SUS PODERES MAGICOS":PRINT :PRINT
"PARA AYUDARTE.":PRINT
14130 PRINT :PRINT "VE A CASA AHORA PORQUE YA ESTAN":PRINT
:PRINT "PREPARANDO UNA FIESTA, PARA CELEBRAR":PRINT
14140 PRINT "EL DESPERTAR DEL LARGO":PRINT :PRINT "SUENYO."
14150 GOTO 14150
```

```
15000 IF OB(30)=-1 THEN OB(30)=20:GOSUB 606:RETURN
15010 IF OB(35)=-1 THEN OB(35)=20:GOSUB 606:RETURN
15020 IF OB(5)=-1 THEN OB(5)=20:GOSUB 606:RETURN
15030 RETURN
15100 IF OB(31)<>1 THEN RETURN
15110 MO$="PESO DEMASIADO Y ME HUNDO !":GOTO 4500
15200 IF RND(1)>0.3 THEN RETURN
15210 MO$="HE CAIDO EN UNA TRAMPA.":GOTO 4500
16000 IF N=10 AND JUGADOR=8 THEN PRINT "NO LO LOGRO !":GOTO
1080
16010 IF N=11 AND JUGADOR=8 THEN PRINT "NO LO LOGRO !":GOTO
1080
16020 GOSUB 600:GOTO 1080
```


Este juego de aventuras ya no precisa de más referencias. No obstante, aquí se trata de la versión completa. La aventura comenzará realmente, cuando usted haya sido capaz de abandonar el territorio conocido hasta ahora.

Entonces entrará en el mundo oscuro de unas viejas minas de oro, y se decidirá, si desea probar una idea espontánea, perdiendo con ello un tiempo precioso, medido aquí por la duración del encendido de una lámpara, o si prefiere asegurarse y realizar solamente aquellas acciones que están justificadas por largas reflexiones.

```
1 REM FIEBRE DEL ORO, VERSION 1.0;ATAR
I
2 REM (C) WALKOWIAK, OCTUBRE 1984
3 REM ----------------------------
10 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
4,1,0:FOR I=1 TO 9:PRINT :NEXT I
20 PRINT "
  "
30 PRINT "     F I E B R E   D E L   O
 R O"
40 PRINT "
  "
60 PRINT "            VERSION 1.0
  ":PRINT
70 PRINT "       (C) 1984 BY WALKOWIAK
  ";
80 FOR I=1 TO 3000
90 NEXT I
100 REM ********* DATOS DE LA AVENTURA
110 AR=31:AO=44:AF=7:JUGADOR=1:WMAX=60
:VALORACION=0:JUGADA=0:IMAX=4:LM=0:L1=
-1:LW=1:L1=20
150 DIM LUGAR$(40),PASADA(AR,6),DIRECC
ION$(36),ENTRADA$(20),T$(40),OB(AO),BL
ANCOS$(38),DZ$(50),ZENDE$(3),FL(AF)
160 DIM VERBOS$(130),EVERBO$(20),EOBJE
TO$(20),OBJETOS$(180),M0$(40),M1$(30),
M2$(30),M3$(30),M4$(38),M5$(38)
170 VERBOS$="EXAMINA   COGE      PON
    ABRE      UTILIZA   DESTRUYE  ENC
IENDE  LLENA     ENTRA     APAGA     "
171 VERBOS$(LEN(VERBOS$)+1)="SUJETA
  "
180 OBJETOS$="ARBOARBOROCACABABOTEMIEL
CAJAESTAEXPLAGUJARCAPLATCUEVOSO ARBUBA
RRPEPIBARRVAGORAILCUERPICOFAROFINA"
181 OBJETOS$(LEN(OBJETOS$)+1)="GANCPOZ
UTABLCASCSACOMUROLIQUCADATERRMONETELAP
ARERDTUORO ORILLAGOPIEDMECHENCEPARA"
190 GOTO 600
```

```
200 REM ******* DESCRIPCIONES DE LUGAR
201 PRINT "EN EL BOSQUE.":RETURN
202 PRINT "EN EL BOSQUE.":RETURN
203 PRINT "EN EL BOSQUE DELANTE DE UN
DESPENADERO":RETURN
204 PRINT "EN UN CLARO.":RETURN
205 PRINT "EN UN CLARO.":RETURN
206 PRINT "EN LA BOCA DE ENTRADA A":PR
INT "UNA VIEJA MINA.":RETURN
207 PRINT "EN LA BOCA DE ENTRADA.":RET
URN
208 PRINT "EN UN HUECO DE LA GALERIA."
:RETURN
209 PRINT "AL FINAL DE LA GALERIA.":RE
TURN
210 PRINT "EN UNA GALERIA LATERAL.":RE
TURN
211 PRINT "EN UNA VIEJA GALERIA DE EXT
RACCION.":RETURN
212 PRINT "EN LA CUEVA.":RETURN
213 PRINT "EN LA CUEVA.":RETURN
214 PRINT "EN LA CUEVA.":RETURN
215 PRINT "EN LA CUEVA.":RETURN
216 PRINT "EN UNA GALERIA.":RETURN
217 PRINT "EN UNA CUEVA LATERAL.":RETU
RN
218 PRINT "EN EL FONDO DEL POZO.":RETU
RN
219 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
220 PRINT "EN UNA GALERIA ANCHA.":RETU
RN
221 PRINT "EN UNA VIEJA GALERIA DE EXT
RACCION.":RETURN
222 PRINT "EN UNA MOLE DE PIEDRA.":RET
URN
223 PRINT "EN UN PARAISO":PRINT "SUBTE
RRANEO.":RETURN
224 PRINT "EN UNA GALERIA SERPENTEADA.
':RETURN
```



```
225 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
226 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
227 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
228 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
229 PRINT "EN UNA GALERIA SERPENTEADA.
":RETURN
230 PRINT "DELANTE DE UN LAGO SUBTERRA
NEO.":RETURN
231 PRINT "EN UNA PEQUENA CUEVA.":RETU
RN
300 REM ********************** OBJETOS
S
301 T$="MUCHOS ARBOLES GRANDES":RETURN

302 T$="MUCHOS ARBOLES GRANDES":RETURN

303 T$="ALGUNAS ROCAS":RETURN
304 T$="UNA CABANA DE MADERA EN RUINAS
":RETURN
305 T$="UNA SUCIA BOTELLA FORRADA":RET
URN
306 T$="MIEL":RETURN
307 T$="UNA CAJA DE MADERA":RETURN
308 T$="UNA ESTANTERIA DESVENCIJADA":R
ETURN
309 T$="UN POCO DE EXPLOSIVO":RETURN
310 T$="UN AGUJERO OSCURO":RETURN
311 T$="UNA OXIDADA ARCA DE HIERRO":RE
TURN
312 T$="*MONEDAS DE PLATA*":RETURN
313 T$="UNA CUEVA OSCURA":RETURN
314 T$="UN OSO FEROZ":RETURN
```

```
315 T$="ARBUSTOS BAJOS":RETURN
316 T$="VARIAS BARRAS DE HIERRO":RETUR
N
317 T$="*PEPITAS DE ORO*":RETURN
318 T$="UNA BARRA DE HIERRO":RETURN
319 T$="UNA VIEJA VAGONETA":RETURN
320 T$="RAILES OXIDADOS":RETURN
321 T$="UNA CUERDA":RETURN
322 T$="UN PICO PESADO":RETURN
323 T$="UN VIEJO FAROL":RETURN
324 T$="EL FINAL DE LA GALERIA":RETURN

325 T$="GANCHOS DE HIERRO":RETURN
326 T$="UN POZO":RETURN
327 T$="UNA PARED DE TABLAS":RETURN
328 T$="CASCOTES":RETURN
329 T$="SACOS VIEJOS":RETURN
330 T$="MUROS DE PIEDRA HUMEDOS":RETUR
N
331 T$="LIQUIDO PESTILENTE":RETURN
332 T$="UN CADAVER":RETURN
333 T$="*TERRONES DE PLATA*":RETURN
334 T$="*MONEDAS DE ORO*":RETURN
335 T$="TELARANAS":RETURN
336 T$="UNA PARED DE ORO":RETURN
337 T$="UN ROTULO":RETURN
338 T$="*ORO*":RETURN
339 T$="UNA ORILLA":RETURN
340 T$="EL LAGO":RETURN
341 T$="PIEDRAS DE ORO":RETURN
342 T$="UNA MECHA":RETURN
343 T$="UN ENCENDEDOR":RETURN
344 T$="":RETURN
400 REM **** POSICIONES OBJETOS *****
410 DATA 1,2,2,3,0,0,3,0,0,0,0,0,5,0,4
,6,5,0,7,7,0,8,8,9,0,9,10,11,0,13,17,1
8,0,0,20,22,22,31,30,30,0,0,-1,0
500 REM ***** TABLA DE DIRECCIONES **
501 DATA 1,1,1,2,0,0,2,1,1,3,0,0,0,4,2
,0,0,0,3,0,5,0,0,0,0,0,6,4,0,0,7,0,1,5
```



```
,0,0,8,6,0,0,0,0,9,7,8,10,0,0
502 DATA 0,8,0,0,0,0,11,0,8,0,0,0,0,10
,0,0,0,0,0,5,0,13,0,0,0,0,12,14,0,0,0,
16,13,15,0,0,0,0,14,0,0,0
503 DATA 14,0,0,17,0,0,0,0,16,0,0,0,0,
0,19,0,0,0,24,0,20,18,0,0,0,0,21,19,0,
0,22,11,11,20,11,0,0,21,0,0,27,0
504 DATA 0,0,0,0,0,0,26,19,24,24,24,24
,27,24,24,29,26,24,27,24,26,27,27,24,2
5,24,26,27,26,24,24,26,26,27,0,0
505 DATA 0,27,30,27,0,0,28,0,0,0,0,0,3
0,0,0,0,0,0
600 REM ********************** MENSAJES
601 M1$="NO VEO NADA ESPECIAL."
602 M2$="NO TENGO TANTA FUERZA !"
603 M3$="COMO TE LO IMAGINAS ?"
604 M4$="EL OSO COGE LA MIEL Y DESAPAR
ECE EN"
605 M5$="EL FONDO DE LA CUEVA."
699 REM ******* TITULO 2: INTRODUCCION
700 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
 4,1,0
710 PRINT "BIENVENIDO A: FIEBRE DEL OR
O":PRINT "UNA AVENTURA PARA ATARI 800
XL":?
720 FOR I=1 TO 37:PRINT CHR$(13);:NEXT
 I:PRINT
730 PRINT "Hace unos dias, y mientras
probaba":PRINT "suerte en un mundo nue
vo, Ud encontro"
740 PRINT "a un viejo moribundo, al qu
e atendio":PRINT "en las ultimas horas
 de su vida."
750 PRINT "Agradecido por el acto, el
viejo le":PRINT "hablo de una mina de
oro que poseia"
760 PRINT "y de parte de la fortuna qu
e alli":PRINT "tenia escondida."
770 PRINT "Ud. ha superado infinidad d
e peligros":PRINT "hasta dar con ella;
```

```
 pronto alcanzara"
780 PRINT "su objetivo y podra comprob
ar, si lo":PRINT "que el viejo decia e
ra cierto o si"
785 PRINT "solo se trataba de un sueno
 delirante":PRINT
790 FOR I=1 TO 37:PRINT CHR$(13);:NEXT
 I:PRINT
795 PRINT "DESEA RECIBIR CONSEJOS PARA
 CONTINUAR";:INPUT ENTRADA$
798 IF ENTRADA$(1,1)="S" THEN GOSUB 80
0
799 GOTO 900
800 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
 4,1,0
820 PRINT "            ATARI - VENTURES
        ":PRINT "      (C) 1984 BY JU
ERG WALKOWIAK      ":REM INVERS
830 PRINT :PRINT "Imaginese un robot a
l que puede":PRINT "controlar con nume
rosos comandos."
840 PRINT "Yo soy este robot y me pond
re en su":PRINT "lugar para enfrentarm
e a todos los"
850 PRINT "peligros de las mas excitan
tes":PRINT "aventuras."
855 PRINT "Para que pueda manejarme co
modamente,":PRINT "le describire en ca
da momento la"
860 PRINT "situacion en que me encuent
re.":PRINT "A continuacion, Ud me indi
ca con dos"
863 PRINT "palabras, como por ejemplo,
":PRINT "EXAMINA PUERTA, COGE CUCHILLO
,"
870 PRINT "lo que yo deba hacer.":PRIN
T "Ademas entiendo las ordenes:":PRINT

880 PRINT "   INSTRUC,  AYUDA,  VOCAB,
":PRINT
```



```
881 PRINT " PUNTOS, INVENT,  SAVE,  LO
AD,  FIN":PRINT
890 PRINT "POR FAVOR, PULSE <RETURN> Y
 ...";:INPUT ENTRADA$:GRAPHICS 0:RETUR
N :REM INVERS
900 FOR I=1 TO AO
910 READ PARADERO:OB(I)=PARADERO
920 NEXT I
930 FOR I=1 TO AF:FL(I)=0:NEXT I
940 FOR LUGAR=1 TO AR
950 FOR DIRECCION=1 TO 6
960 READ META:PASADA(LUGAR,DIRECCION)=
META
970 NEXT DIRECCION
980 NEXT LUGAR
1000 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLO
R 4,1,0:TRAP 1080:GOTO 1020
1010 REM **** TEXTO CLARO: DIRECCIONES
1011 T$="N (NORTE), ":RETURN
1012 T$="S (SUR), ":RETURN
1013 T$="O (OESTE), ":RETURN
1014 T$="E (ESTE), ":RETURN
1015 T$="AR (ARRIBA), ":RETURN
1016 T$="AB (ABAJO), ":RETURN
1020 BLANCOS$="
                "
1030 T$=""
1040 ZENDE$=CHR$(30):ZENDE$(LEN(ZENDE$
)+1)=CHR$(30):ZENDE$(LEN(ZENDE$)+1)=".
"
1080 PRINT :JUGADA=JUGADA+1:LW=LW+1
1084 IF LW=LM THEN GOSUB 3000
1085 IF VALORACION=WMAX THEN GOTO 4800
1090 FOR LINEA=0 TO 10:POSITION 2,LINE
A:PRINT BLANCOS$:NEXT LINEA
1091 REM ********************** SIN LUZ
1092 IF LI=-1 THEN 1100
1093 PRINT "NO SE EXACTAMENTE DONDE ES
TOY.":PRINT "ES DEMASIADO OSCURO PARA
VER ALGO."
```

```
1095 PRINT :PRINT "TAMPOCO PUEDO VER L
AS BOCAS DE":PRINT "SALIDA !":GOTO 132
0
1100 POSITION 2,0:PRINT "ESTOY ";:GOSU
B 200+JUGADOR
1110 DZ$="VEO ":IMPRIMIDO=0
1120 FOR I=1 TO AO
1130 IF OB(I)<>JUGADOR THEN GOTO 1170
1140 IMPRIMIDO=-1:GOSUB 300+I:T$(LEN(T
$)+1)=", "
1150 IF LEN(DZ$)+LEN(T$)>=38 THEN PRIN
T DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 117
0
1160 DZ$(LEN(DZ$)+1)=T$:T$=""
1170 NEXT I
1180 IF  NOT IMPRIMIDO THEN DZ$(LEN(DZ
$)+1)="NADA ESPECIAL."
1190 DZ$(LEN(DZ$)+1)=ZENDE$
1200 PRINT DZ$
1210 PRINT BLANCOS$
1220 DZ$="PUEDO IR HACIA ":IMPRIMIDO=0
1230 FOR DIRECCION=1 TO 6
1240 IF PASADA(JUGADOR,DIRECCION)=0 TH
EN GOTO 1280
1250 GOSUB 1010+DIRECCION:IMPRIMIDO=-1
1260 IF LEN(DZ$)+LEN(T$)>=38 THEN PRIN
T DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 128
0
1270 DZ$(LEN(DZ$)+1)=T$:T$=""
1280 NEXT DIRECCION
1290 IF  NOT IMPRIMIDO THEN DZ$(LEN(DZ
$)+1)="EN NINGUN SITIO. "
1300 DZ$(LEN(DZ$)+1)=ZENDE$
1310 PRINT DZ$
1320 PRINT "RRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRR":REM 37 CONTROL R
1340 IF VALORACION=WMAX THEN GOTO 4800
1350 IF JUGADOR=15 THEN MO$="DESGRACIA
DAMENTE ALLI ESTABA EL OSO.":FOR I=1 T
O 600:NEXT I:GOTO 4500
```



```
1360 IF LW<=0 THEN LI=0
1370 IF EX=JUGADA AND JUGADOR=20 THEN
MO$="RUMS! - A MI TAMBIEN ME HA DESTRU
ZADO":GOTO 4500
1380 IF JUGADOR>18 AND OB(23)<>-1 THEN
 LI=0
1390 POSITION 2,23:PRINT "QUE DEBO HAC
ER ";:INPUT ENTRADA$
1395 RL=LM-LW:IF (RL>0 AND RL<15) THEN
 PRINT "EN";LM-LW;"MOVIMIENTOS ESTARE
A OSCURAS!"
1400 IF LEN(ENTRADA$)>2 THEN 1480
1410 IF ENTRADA$="N" AND PASADA(JUGADO
R,1)<>0 THEN JUGADOR=PASADA(JUGADOR,1)
:PRINT "VALE":GOTO 1080
1420 IF ENTRADA$="S" AND PASADA(JUGADO
R,2)<>0 THEN JUGADOR=PASADA(JUGADOR,2)
:PRINT "VALE":GOTO 1080
1430 IF ENTRADA$="O" AND PASADA(JUGADO
R,3)<>0 THEN JUGADOR=PASADA(JUGADOR,3)
:PRINT "VALE":GOTO 1080
1440 IF ENTRADA$="E" AND PASADA(JUGADO
R,4)<>0 THEN JUGADOR=PASADA(JUGADOR,4)
:PRINT "VALE":GOTO 1080
1450 IF ENTRADA$="AR" AND PASADA(JUGAD
OR,5)<>0 THEN JUGADOR=PASADA(JUGADOR,5
):PRINT "VALE":GOTO 1080
1460 IF ENTRADA$="AB" AND PASADA(JUGAD
OR,6)<>0 THEN JUGADOR=PASADA(JUGADOR,6
):PRINT "VALE":GOTO 1080
1470 IF LEN(ENTRADA$)<3 THEN PRINT "HA
CIA ALLI NO CONDUCE NINGUN CAMINO!":GO
TO 1080
1480 IF LEN(ENTRADA$)>8 THEN GOTO 2000
1499 REM ********** COMIENZO INVENTARIO
1500 IF ENTRADA$(1,3)<>"INV" THEN GOTO
 1560
1510 PRINT "LLEVO CONMIGO LO SIGUIENTE
: "
1520 FOR I=1 TO AO
```

```
1530 IF OB(I)=-1 THEN GOSUB 300+I:PRIN
T T$
1540 NEXT I
1550 GOTO 1080
1551 REM ********** FIN INVENTARIO
1560 IF ENTRADA$(1,3)<>"PUN" THEN GOTO
 1600
1561 PRINT "DE ";WMAX;" PUNTOS HAS OBT
ENIDO ";VALORACION;:PRINT " PUNTOS EN
";JUGADA;" JUGADAS!"
1563 PRINT "ESTO ES, UNA MEDIA DE ";VA
LOR/JUGADA;" PUNTOS.":GOTO 1080
1599 REM ********** SAVE GAME
1600 IF ENTRADA$(1,3)<>"SAV" THEN GOTO
 1700
1605 PRINT "CON QUE NOMBRE ";:INPUT EN
TRADA$:IF LEN(ENTRADA$)>8 THEN PRINT "
UN OCO MAS CORTO, POR FAVOR!":GOTO 160
5
1610 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)
=".DAT"
1615 T$(LEN(T$)+1)=ENTRADA$
1620 OPEN #1,8,0,T$
1625 PRINT #1,JUGADOR
1630 FOR I=1 TO AO
1631 PRINT #1,OB(I)
1632 NEXT I
1635 FOR LUGAR=1 TO AR
1636 FOR DIRECCION=1 TO 6
1637 PRINT #1,PASADA(LUGAR,DIRECCION)
1638 NEXT DIRECCION
1639 NEXT LUGAR
1645 FOR I=1 TO AF
1646 PRINT #1,FL(I)
1647 NEXT I
1650 CLOSE #1
1670 PRINT "VALE":GOTO 1080
1699 REM ********** LOAD GAME
1700 IF ENTRADA$(1,3)<>"LOA" THEN GOTO
 1800
```



```
1705 PRINT "QUE JUEGO ";:INPUT ENTRADA
$:IF LEN(ENTRADA$)>8 THEN PRINT "ESTE
NO PUEDE EXISTIR!":GOTO 1705
1710 T$="D:":ENTRADA$(LEN(ENTRADA$)+1)
=".DAT"
1715 T$(LEN(T$)+1)=ENTRADA$
1720 OPEN #1,4,0,T$
1725 INPUT #1,JUGADOR
1730 FOR I=1 TO AO
1731 INPUT #1,PARADERO:OB(I)=PARADERO
1732 NEXT I
1735 FOR LUGAR=1 TO AR
1736 FOR DIRECCION=1 TO 6
1737 INPUT #1,META:PASADA(LUGAR,DIRECC
ION)=META
1738 NEXT DIRECCION
1739 NEXT LUGAR
1745 FOR I=1 TO AF
1746 INPUT #1,FLAG:FL(I)=FLAG
1747 NEXT I
1750 CLOSE #1
1770 PRINT "VALE":GOTO 1080
1800 IF ENTRADA$(1,3)<>"VOC" THEN GOTO
 1900
1810 GOSUB 1880:IMPRIMIDO=0
1830 FOR I=1 TO LEN(VERBOS$) STEP 10
1840 PRINT VERBOS$(I,I+9):IMPRIMIDO=IM
PRIMIDO+1
1850 IF IMPRIMIDO=18 THEN GOSUB 1890:I
MPRIMIDO=1:GOSUB 1880
1860 NEXT I
1870 GOSUB 1890:GRAPHICS 0:SETCOLOR 2,
1,0:SETCOLOR 4,1,0:GOTO 1080
1880 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLO
R 4,1,0:PRINT "CONOZCO LAS ACCIONES SI
GUIENTES ...":PRINT :RETURN
1890 PRINT :PRINT :PRINT "PULSAR <RETU
RN> ...":INPUT ENTRADA$:RETURN
1900 IF ENTRADA$(1,3)<>"INS" THEN GOTO
 1950
```

```
1910 GOSUB 800
1920 GOTO 1080
1950 IF ENTRADA$(1,3)<>"FIN" THEN GOTO
 1970
1960 GRAPHICS 0:PRINT "EL AUTOR LE DES
EA MAS SUERTE EN UN":PRINT "EN UN PROX
IMO INTENTO!":PRINT :PRINT :PRINT :END

1970 IF ENTRADA$(1,3)<>"AYU" THEN GOTO
 2000
1971 IF JUGADOR=4 AND OB(10)=0 THEN PR
INT "CASI ME HUBIERA CAIDO EN UN FOSO"
:GOTO 1080
1972 IF JUGADOR=4 AND OB(11)<>JUGADOR
AND  NOT FL(2) THEN PRINT "NECESITO AL
GO PARA ROMPER LA CADENA":GOTO 1080
1975 PRINT "PRIMERO MIRAR , DESPUES PE
NSAR Y AL":PRINT "FINAL ACTUAR !!":GOT
O 1080
1979 REM ********************* FIN HELP
1999 REM ************ SEPARAR ENTRADA
2000 LN=LEN(ENTRADA$)
2010 FOR I=1 TO LN
2020 IF ENTRADA$(I,1)<>" " THEN NEXT I
2030 EVERBO$=ENTRADA$(1,I)
2040 IF LEN(EVERBO$)=LN-1 THEN GOTO 20
90
2050 EOBJETO$=ENTRADA$(I+1,LN)
2060 VN=0:N=0
2070 EVERBO$=EVERBO$(1,3)
2080 EOBJETO$=EOBJETO$(1,3)
2090 FOR I=1 TO LEN(VERBOS$)-2 STEP 10
2100 VN=VN+1
2110 IF VERBOS$(I,I+2)=EVERBO$ THEN 21
40
2120 NEXT I
2130 PRINT "NO ENTIENDO EL VERBO!":GOT
O 1080
2140 FOR I=1 TO LEN(OBJETOS$) STEP 4
2150 N=N+1
```



```
2160 IF OBJETOS$(1,I+2)=EOBJETO$ THEN
GOTO 2200
2170 NEXT I
2180 PRINT "NO ENTIENDO EL OBJETO!":GO
TO 1080
2190 REM INS    COG   PON   ABRUTI   D
ES
2200 ON VN GOTO 5000,2210,7000,8000,90
00,10000,11000,12000,13000,14000,15000
2209 REM ********* LIMITAR INVENTARIO
2210 ANZ=0:FOR I=1 TO A0
2220 IF OB(I)=-1 THEN ANZ=ANZ+1
2230 IF ANZ=IMAX THEN PRINT "NO GRACIA
S.- YA LLEVO BASTANTE CARGA.":GOTO 108
0
2240 NEXT I
2250 GOTO 6000
2260 REM ***************** FIN INVTEST
2999 REM ************** UP INTERRUPTOR
3000 IF LI=-1 THEN LI=0:GOTO 3020
3010 IF LI=0 THEN LI=-1
3020 LW=0:RETURN
4500 GRAPHICS 0:REM *** JUGADOR MUERTO
4600 PRINT "LO QUE FALTABA!":PRINT :PR
INT MO$
4610 PRINT :PRINT "ESTOY MUERTO!":PRIN
T
4620 PRINT "QUIERE QUE LO INTENTE OTRA
 VEZ ";:INPUT ENTRADA$
4630 IF ENTRADA$(1,1)="S" THEN CLR :GO
TO 100
4640 GOTO 1960
4800 GRAPHICS 0
4810 PRINT "ENHORABUENA!"
4820 PRINT :PRINT :PRINT "HA RESUELTO
EL PROBLEMA PLANTEADO":PRINT
4830 PRINT "Y PUEDE AFRONTAR UNA NUEVA
 AVENTURA."
4840 PRINT :PRINT :PRINT :END
4900 PRINT "ATENCION ERROR!":GOTO 1080
```

```
4999 REM **EJECUTAR MOVIMIENTOS JUGADO
R
5000 IF OB(N)<>JUGADOR AND OB(N)<>-1 T
HEN GOTO 5900
5002 IF N=1 THEN PRINT M1$:GOTO 1080
5003 IF N=3 THEN PRINT M1$:GOTO 1080
5004 IF N=4 THEN PRINT "EN UN RINCON H
AY UNA ESTANTERIA.":OB(8)=JUGADOR:GOTO
 1080
5005 IF N=5 THEN PRINT "LA BOTELLA EST
A LLENA DE MIEL.":GOTO 1080
5006 IF N=6 THEN PRINT M1$:GOTO 1080
5007 IF N=7 AND OB(9)=0 THEN PRINT "EN
 LA CAJA HAY EXPLOSIVOS.":GOTO 1080
5008 IF N=8 AND OB(5)=0 THEN PRINT "EN
 LA ESTANTERIA HAY UNA BOTELLA":PRINT
"FORRADA.":OB(5)=JUGADOR:GOTO 1080
5009 IF N=8 AND OB(5)<>0 THEN PRINT M1
$:GOTO 1080
5010 IF N=10 THEN PRINT "EN EL AGUJERO
 HAY UN ARCA DE HIERRO.":OB(11)=JUGADO
R:GOTO 1080
5011 IF N=11 AND  NOT FL(1) THEN PRINT
 "ESTA CERRADA CON UNA CADENA.":GOTO 1
080
5012 IF N=11 AND FL(1) AND  NOT FL(2)
THEN PRINT "POR FUERA NO VEO NADA ESPE
CIAL.":GOTO 1080
5013 IF N=11 AND FL(1) AND FL(2) AND O
B(12)=0 THEN PRINT "ESTA LLENA DE MONE
DAS DE PLATA.":OB(12)=JUGADOR:GOTO 108
0
5014 IF N=12 THEN PRINT "ES JUSTO LO Q
UE BUSCABA!!":GOTO 1080
5015 IF N=13 THEN PRINT "HE ASUSTADO A
 UN OSO.":OB(14)=JUGADOR:GOTO 1080
5017 IF N=15 THEN PRINT "EN MEDIO HAY
UN AGUJERO.":OB(10)=JUGADOR:GOTO 1080
5018 IF N=16 THEN PRINT "PARECEN SER M
UY RESISTENTES.":GOTO 1080
```



```
5019 IF N=17 THEN PRINT "SE TRATA DE O
RO PURO!":GOTO 1080
5020 IF N=19 AND OB(21)=0 THEN PRINT "
TODAVIA PENDE LA CUERDA DE TIRO.":OB(2
1)=JUGADOR:GOTO 1080
5021 IF N=23 THEN PRINT "ES UNA VIEJA
LAMPARA DE ACEITE":GOTO 1080
5022 IF N=24 THEN PRINT "SALE UN GANCH
O DE LA PARED.":OB(25)=JUGADOR:GOTO 10
80
5023 IF N=25 THEN PRINT "LO SIENTO - E
STA COGIDO MUY FUERTE.":GOTO 1080
5024 IF N=26 THEN MO$="ES MUY HONDO -
Y YO ESTABA DEMASIADO  CERCA DEL BORDE
.":GOTO 4500
5025 IF N=27 THEN PRINT "SU CARPINTERO
 ERA UN ARTISTA.":GOTO 1080
5026 IF N=28 AND OB(N)=0 THEN PRINT "H
E DESCUBIERTO DOS SACOS.":OB(29)=JUGAD
OR:GOTO 1080
5027 IF N=29 THEN PRINT "ESTAN LLENOS
DE ORO EN POLVO.":GOTO 1080
5028 IF N=31 THEN PRINT "ES VISCOSA, O
LEOSA Y SE PEGA EN LOS":PRINT "DEDOS."
:GOTO 1080
5029 IF N=32 THEN PRINT "APESTA!"
5030 IF N=32 AND OB(34)=0 THEN PRINT "
EN EL BOLSILLO HAY MONEDAS DE ORO.":OB
(34)=JUGADOR:GOTO 1080
5031 IF N=35 THEN PRINT "NO SON TELARA
NAS.":OB(N)=0:OB(42)=JUGADOR
5032 IF N=35 THEN PRINT "ES UNA MECHA.
":GOTO 1080
5033 IF N=36 THEN PRINT "REALMENTE ES
ORO PURO.":GOTO 1080
5034 IF N=37 THEN PRINT "LLEVA ESCRITO
:":PRINT "UNA VEZ DESPIERTA LA CODICIA
,":? "TAMBIEN LA MUERTE SE BENEFICIA" ,
5035 IF N=37 THEN GOTO 1080
5036 IF N=42 THEN PRINT "CONDUCE HACIA
```

```
 EL TECHO.":GOTO 1080
5037 IF N=43 THEN PRINT "ES UN TIPO DE
 ENCENDEDOR HABITUAL EN":PRINT "EL OES
TE."
5038 IF N=32 AND FL(4) AND OB(33)<>-2
THEN PRINT "ESTABA ENCIMA DE UNOS TROZ
OS DE PLATA":OB(33)=JUGADOR:GOTO 1080
5899 PRINT M1$:GOTO 1080
5900 REM ********** OBJETO NO PRESENTE
5901 IF N=1 AND JUGADOR=2 THEN PRINT M
1$:GOTO 1080
5902 IF N=6 AND OB(5)=-1 THEN PRINT "E
S DULCE Y SABROSA.":GOTO 1080
5903 IF N=6 AND OB(5)<>-1 THEN PRINT "
NO TENGO MIEL!":GOTO 1080
5904 IF (N=9 AND (OB(7)=JUGADOR OR OB(
7)=-1)) THEN PRINT "PARECE SER MUY EXP
LOSIVO!":GOTO 1080
5910 IF N=14 AND JUGADOR=5 AND OB(14)=
5 THEN PRINT "PARECE QUE YO DESPIERTE
SU APETITO.":GOTO 1080
5911 IF N=20 AND JUGADOR=22 THEN ? "UN
A VEZ DESPIERTA LA CODICIA,":? "TAMBIE
N LA MUERTE SE BENEFICIA.":GOTO 1080
5912 IF N=44 AND JUGADOR=23 THEN MO$="
ESTOY EN EL REINO DE LOS MUERTOS.":GOT
O 4500
5990 PRINT "AQUI NO VEO NADA PARECIDO!
":GOTO 1080
6000 IF OB(N)<>JUGADOR AND OB(N)<>-1 T
HEN GOTO 6900
6001 IF N=1 THEN PRINT M2$:GOTO 1080
6002 IF N=3 THEN PRINT M2$:GOTO 1080
6003 IF N=4 THEN PRINT M3$:GOTO 1080
6004 IF N=8 THEN PRINT M2$:GOTO 1080
6005 IF N=11 THEN PRINT M2$:GOTO 1080
6006 IF N=10 THEN PRINT M3$:GOTO 1080
6007 IF N=13 THEN PRINT M3$:GOTO 1080
6008 IF N=15 THEN PRINT M2$:GOTO 1080
6010 IF N=5 THEN OB(5)=-1:PRINT "VALE"
```



```
:GOTO 1080
6011 IF N=6 THEN OB(5)=-1:PRINT "VALE"
:GOTO 1080
6012 IF N=7 THEN OB(7)=-1:PRINT "VALE"
:GOTO 1080
6014 IF N=12 THEN OB(12)=-2:PRINT "VAL
E":VALORACION=VALORACION+10:GOTO 1080
6015 IF N=1 AND JUGADOR=5 THEN MO$="EL
 OSO ME HA MATADO.":GOTO 4500
6016 IF N=16 THEN OB(18)=-1:PRINT "VAL
E":GOTO 1080
6017 IF N=17 AND FL(3) THEN PRINT "O.K
.":OB(17)=-2:VALORACION=VALORACION+10:
GOTO 1080
6018 IF N=17 AND  NOT FL(3) THEN MO$="
UN OSO SE LANZA SOBRE MI.":GOTO 4500
6019 IF N=12 AND OB(N)=-2 THEN PRINT "
YA TENGO LA PLATA.":GOTO 1080
6020 IF N=17 AND OB(N)=-2 THEN PRINT "
YA TENGO EL ORO.":GOTO 1080
6021 IF N=19 THEN PRINT M2$:GOTO 1080
6022 IF N=21 AND FL(4) THEN PRINT "VAL
E":OB(N)=-1:GOTO 1080
6023 IF N=22 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6024 IF N=24 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6025 IF N=25 THEN PRINT "ESTA CLAVADO
DEMASIADO FUERTE EN LA":PRINT "ROCA.":
GOTO 1080
6026 IF N=27 THEN PRINT M3$:GOTO 1080
6027 IF N=28 THEN PRINT "Y QUE HAGO CO
N ESTO?":GOTO 1080
6030 IF N=31 AND OB(5)<>-1 THEN PRINT
"CON QUE?":GOTO 1080
6031 IF N=31 AND OB(5)=-1 THEN PRINT "
O.K. - LA BOTELLA ESTA LLENA.":FL(6)=-
1:GOTO 1080
6032 IF N=32 THEN PRINT "VALE - OH, QU
E ES ESTO?":FL(5)=-1:GOTO 1080
```

```
6033 IF N=33 AND OB(N)=JUGADOR THEN PR
INT "VALE":OB(N)=-2:VALORACION=VALORAC
ION+10:GOTO 1080
6034 IF N=34 AND OB(N)=JUGADOR THEN PR
INT "VALE":OB(N)=-2:VALORACION=VALORAC
ION+10:GOTO 1080
6035 IF N=35 THEN PRINT "NO SON TELARA
NAS, ERA UNA MECHA.":OB(N)=0:OB(42)=-1
:GOTO 1080
6037 IF N=37 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6038 IF N=41 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6039 IF N=42 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6040 IF N=43 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6041 IF N=21 AND  NOT FL(4) THEN PRINT
 "ESTA ATADO A LA VAGONETA.":GOTO 1080
6042 IF N=23 THEN PRINT "VALE":OB(N)=-
1:GOTO 1080
6043 IF N=29 AND OB(N)=JUGADOR THEN PR
INT "VALE":OB(N)=-2:VALORACION=VALORAC
ION+10:GOTO 1080
6044 IF N=38 AND OB(N)=JUGADOR THEN PR
INT "VALE":OB(N)=-2:VALORACION=VALORAC
ION+10:GOTO 1080
6900 IF N=9 AND (OB(7)=JUGADOR OR OB(7
)=-1) THEN MO$="EL EXPLOSIVO HA ESTALL
ADO AL CONTACTO":GOTO 4500
6910 IF N=20 AND JUGADOR=22 THEN PRINT
 "UNA VEZ DESPIERTA LA CODICIA,":PRINT
 "TAMBIEN LA MUERTE SE BENEFICIA."
6915 GOTO 1080
6999 PRINT "AQUI NO VEO NADA PARECIDO!
":GOTO 1080
7000 IF N=16 AND OB(18)=-1 THEN PRINT
"VALE":OB(18)=JUGADOR:GOTO 1080
7005 IF OB(N)<>-1 THEN PRINT "PERO SI
NO TENGO NADA DE ESTO!":GOTO 1080
```



```
7010 IF N=6 AND JUGADOR=5 THEN OB(6)=0
:FL(3)=-1:PRINT M4$:PRINT M5$:OB(14)=0
:GOTO 1080
7020 IF N=5 AND JUGADOR=5 THEN OB(5)=0
:FL(3)=-1:PRINT M4$:PRINT M5$:OB(14)=0
:GOTO 1080
7900 OB(N)=JUGADOR:PRINT "VALE":GOTO 1
080
8000 IF OB(N)<>JUGADOR AND OB(N)<>-1 T
HEN PRINT "AQUI NO VEO NADA PARECIDO."
:GOTO 1080
8005 IF N=4 AND JUGADOR=3 THEN PRINT "
LA CABANA YA ESTABA ABIERTA.":GOTO 108
0
8010 IF N=5 THEN PRINT "VALE":GOTO 108
0
8020 IF N=11 AND  NOT FL(1) THEN PRINT
 "LA CADENA NO LO PERMITE!":GOTO 1080
8025 IF N=11 AND FL(1) THEN PRINT "VAL
E - LA TAPA CAE HACIA ATRAS.":FL(2)=-1
:GOTO 1080
8030 IF N=23 THEN PRINT "VALE":GOTO 10
80
8035 IF N=29 THEN PRINT "VALE":GOTO 10
80
8040 IF N=32 THEN PRINT "LO SIENTO!":P
RINT "NO SOY FRANKENSTEIN.":GOTO 1080
8045 IF N=36 THEN PRINT "COMO?":GOTO 1
080
8999 PRINT "NO ENTIENDO LO QUE QUIERES
 DECIR.":GOTO 1080
9000 IF OB(N)<>JUGADOR AND OB(N)<>-1 T
HEN GOTO 9900
9010 IF N=16 AND JUGADOR=4 THEN PRINT
"LA CADENA SE ROMPE.":FL(1)=-1:GOTO 10
80
9020 IF N=19 THEN PRINT "COMO Y PARA Q
UE?":GOTO 1080
9030 IF N=21 THEN PRINT "COMO Y PARA Q
UE?":GOTO 1080
```

```
9900 IF N=16 AND OB(18)=-1 AND JUGADOR
=4 THEN PRINT "LA CADENA SE ROMPE.":FL
(1)=-1:GOTO 1080
9999 PRINT "NO ENTIENDO LO QUE QUIERES
 DECIR!":GOTO 1080
10000 IF N=18 AND JUGADOR=4 AND OB(18)
=-1 THEN PRINT "LA CADENA SE ROMPE.":F
L(1)=-1:GOTO 1080
10010 IF N=18 AND JUGADOR=4 AND OB(18)
<>-1 THEN PRINT "NO TENGO NINGUNA BARR
A DE HIERRO.":GOTO 1080
10020 IF N=36 AND OB(22)=-1 THEN PRINT
 "VALE":PASADA(22,1)=23:GOTO 1080
10030 IF N=21 AND (OB(N)=JUGADOR OR OB
(N)=-1) THEN PRINT "VALE":FL(4)=-1:GOT
O 1080
10040 IF N=27 AND OB(22)=-1 THEN PRINT
 "VALE":PASADA(10,2)=12:GOTO 1080
10050 IF N=27 AND OB(22)<>-1 THEN PRIN
T "CON QUE?":GOTO 1080
10999 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR!":GOTO 1080
11000 IF OB(43)<>-1 THEN PRINT "NO TEN
GO NADA PARA ENCENDER.":GOTO 1080
11010 IF N=8 AND LZ<=0 THEN PRINT "NO
QUEDA ACEITE EN LA LAMPARA.":GOTO 1080
11020 IF N=42 AND (OB(43)=-1 OR L1=-1)
 THEN PRINT "ZZZTTSCH",:FL(7)=-1:EX=JU
GADA+3:OB(N)=0
11021 IF N=42 AND (OB(43)=-1 OR L1=-1)
 THEN PASADA(30,2)=31:PASADA(31,1)=30:
GOTO 1080
11030 IF N=23 AND OB(23)=-1 THEN PRINT
 "VALE - LA LAMPARA ESTA ENCENDIDA.":L
M=L1:LW=1:GOTO 1080
11040 IF N=43 THEN PRINT "VALE - LA ME
CHA SE CONSUME.":GOTO 1080
11998 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR.":GOTO 1080
12000 REM ********************** LLENA
```



```
12010 IF N=23 AND (FL(6) AND OB(23)=-1
) THEN LM=60:FL(6)=0:LW=0:PRINT "VALE"
:GOTO 1080
12020 IF N=5 AND JUGADOR=17 THEN FL(6)
=-1:PRINT "LA BOTELLA ESTA LLENA.":GOT
O 1080
12030 IF N=43 AND OB(43)=-1 THEN PRINT
 "LA MECHA SE CONSUME.":GOTO 1080
12998 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR.":GOTO 1080
13000 REM ********************** ENTRA
13010 IF N=13 AND JUGADOR=5 THEN JUGAD
OR=12:PRINT "VALE":GOTO 1080
13998 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR.":GOTO 1080
14000 IF N=23 AND OB(N)=-1 THEN L1=LM:
LW:PRINT "VALE":GOTO 1080
14998 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR.":GOTO 1080
15000 IF N=21 AND JUGADOR=9 THEN PRINT
 "VALE":OB(N)=9:PASADA(9,6)=18:PASADA(
18,5)=9:GOTO 1080
15998 PRINT "NO ENTIENDO LO QUE QUIERE
S DECIR.":GOTO 1080
```

SPACE MISSION

Por favor, no entre este programa llevado por la atracción de una aventura gráfica, a menos que usted sea un aficionado absoluto a esta materia, y que sus escasos conocimientos de programación no le permitan extraer del listado las respuestas a las preguntas que se le harán nada más comenzar el juego.

A diferencia de las demás aventuras, donde las situaciones peligrosas se producen a causa de sus acciones, en Space Mission, usted sólo dispone de nueve (¡9!) jugadas para salvar su vida después de iniciar la partida con RUN.

Sería una lástima que usted perdiese las ganas de jugar con programas de este tipo, sólo porque no pueda sentir la satisfacción de conseguir un éxito.

A tipear 19 pag 14

```
50 GRAPHICS 0:SETCOLOR 2,1,0:SETCOLOR
4,1,0:COLOR 1
55 PRINT "SPACE MISSION":? "(C) 1984 B
Y J.W."
115 AR=30:AO=50:AF=11:SP=11:IMAX=2:ZUG
=0
140 DIM RA$(35),DU(AR,6),RI$(36),EINGA
BE$(20),T$(30),OB(AO),DZ$(50),ZE$(3),F
L(AF),OK$(4),TM$(6)
145 DIM VERBEN$(97),EV$(10),EO$(10),OB
JEKTE$(201),MO$(46):OK$="VALE":TRAP 10
80
155 VERBEN$="EXAMINA   COGE      ABRE
    PULSA      COLOCA    LLEVA      "
160 VERBEN$(LEN(VERBEN$)+1)="LLENA
 PON        UTILIZA    CAMBIA      "
165 OBJEKTE$="CENTVIDETERMKIPETELEANTI
ENTRENTRGEHAORDEEXPAMODUHEXACABLTUBORO
TU"
170 OBJEKTE$(LEN(OBJEKTE$)+1)="LANDARM
AMONIPUERAUTOMANDCATRTISCMOVEDEPOOXIGT
RAJLLAVSTOPATORPUES"
175 OBJEKTE$(LEN(OBJEKTE$)+1)="PUERSET
CMODUMEDIKOORRADIMAGNSCHO*SD*VIEWBFITS
TATSELOTRANKONTPASSCAJA":GOTO 900
201 PRINT "EN LA SALA DE MANDOS.":RETU
RN
202 PRINT "EN LA SALA DE TRANSMISION."
:RETURN
203 PRINT "EN LA CENTRAL EMISORA.":RET
URN
204 PRINT "EN EL PUESTO DEL PILOTO.":R
ETURN
205 PRINT "*":RETURN
206 PRINT "EN EL CONTROL DE ENTRADA.":
RETURN
207 PRINT "EN UN ENTREPUENTE.":RETURN
208 PRINT "EN LA SECCION DE ASTRONOMIA
.":RETURN
209 PRINT "EN LA CUPULA POLAR SUPERIOR
```

```
.":RETURN
210 PRINT "EN LA ESTACION MEDICA.":RET
URN
211 PRINT "EN LA CANTINA.":RETURN
212 PRINT "EN LA SALA DE TRANSMISION."
:RETURN
213 PRINT "EN UNA SALA DE CARGA.":RETU
RN
214 PRINT "EN LA SALA DE MAQUINAS.":RE
TURN
215 PRINT "EN UN ALMACEN.":RETURN
216 PRINT "ANTE UN MAMPARO DE SEGURIDA
D.":RETURN
217 PRINT "EN UN PASILLO.":RETURN
218 PRINT "EN UN PASILLO.":RETURN
219 PRINT "EN UN PASILLO.":RETURN
220 PRINT "EN EL ALMACEN DE REPUESTOS.
":RETURN
221 PRINT "EN UN ENTREPUENTE.":RETURN
222 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
223 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
224 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
225 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
226 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
227 PRINT "EN LA CAJA ANTIGRAVEDAD.":R
ETURN
230 PRINT "EN UN SALA DE TRIPULACION."
:RETURN
301 T$="EL PUESTO DE MANDO CENTRAL.":RE
TURN
302 T$="PANTALLA DE VIDEO":RETURN
303 T$="NADA ESPECIAL":RETURN
304 T$="TERMINAL":RETURN
305 T$="HIPERCOM":RETURN
306 T$="TELECOM":RETURN
```



```
307 T$="":RETURN
308 T$="ENTRADA":RETURN
309 T$="":RETURN
310 T$="":RETURN
311 T$="ORDENADOR":RETURN
312 T$="MODULO DE EXPANSION":RETURN
313 T$="MODULOS":RETURN
314 T$="LLAVE MACHO HEXAGONAL":RETURN
315 T$="ARBOLES DE CABLES":RETURN
316 T$="TUBOS":RETURN
317 T$="ROTULO":RETURN
318 T$="":RETURN
319 T$="ARMARIO":RETURN
320 T$="":RETURN
321 T$="PUERTA":RETURN
322 T$="EL PILOTO AUTOMATICO":RETURN
323 T$="UN BOTON DE MANDO":RETURN
324 T$="UN CATRE":RETURN
325 T$="NADA ESPECIAL":RETURN
326 T$="":RETURN
327 T$="DEPOSITOS":RETURN
328 T$="BOTELLA DE OXIGENO":RETURN
329 T$="UN TRAJE ESPACIAL":RETURN
330 T$="LLAVE DEL 10":RETURN
331 T$="":RETURN
332 T$="ATORNILLADOR":RETURN
333 T$="PUESTO DE MANDO":RETURN
334 T$="UNA PUERTA ROJA":RETURN
335 T$="":RETURN
336 T$="UN MODULO":RETURN
337 T$="MEDICAMENTOS":RETURN
338 T$="":RETURN
339 T$="APARATOS DE RADIO":RETURN
340 T$="TARJETA MAGNETICA":RETURN
341 T$="NADA ESPECIAL.":RETURN
342 T$="":RETURN
343 T$="":RETURN
344 T$="":RETURN
345 T$="":RETURN
346 T$="":RETURN
```

```
347 T$="":RETURN
348 T$="NADA ESPECIAL":RETURN
349 T$="":RETURN
350 T$="CAJAS":RETURN
401 DATA 1,1,30,12,0,0,0,0,0,7,0,-1,0,
0,7,9,8,0,10,0,9,0,9,10,11,0,21,0,0,0,
0,0,14,14,0,0,0,0,3,0,16,0,0,30,0,0,0
402 DATA 6,0,15,3,3,16,0,0,0,6,0,0,0,0
,0,1,1,0,0,0,0,3,0,0,6,0,0,0,0,0,0,0,0
,0,2,0,0,0,0,0,0,23,0,0,0,0,0,22,0,0
403 DATA 0,0,0,0,0,0,0,22,0,0,0,22,0,0,0
,0,30,24,0,0,12,12,18,13,0,0,0,0,12,0,
0,0,0,0,17,0,0,0,0,2,0,0,0,0,0
404 DATA 0,24,0,0,0,19,19,26,14,0,0,18
,18,27,12,0,0,17,17,20,0,0,0,0,0,0,19,
0,0,0,0,25,0,0,0,0,0,10,8,9,23
405 DATA 0,0,0,0,22,24,0,0,11,16,23,0,
0,0,0,0,24,26,0,0,0,17,25,27,0,0,0,18,
26,0,0,0,0,0,0,0,0,0,0,0,0,0,0
406 DATA 0,0,0,11,0,0
601 PRINT "NO VEO NADA ESPECIAL.":RETU
RN
602 PRINT "NO TENGO TANTA FUERZA.":RET
URN
603 PRINT "COMO TE LO IMAGINAS?":RETUR
N
604 PRINT "PERO SI ESTO YA LO TENGO!":
RETURN
605 PRINT "NO ENTIENDO LO QUE QUIERES
DECIR.":RETURN
900 FOR I=1 TO AO:READ LAGEORT:OB(I)=L
AGEORT:NEXT I
910 FOR I=1 TO AF:FL(I)=0:NEXT I
920 FOR RA=1 TO AR:FOR RI=1 TO 6:READ
Z:DU(RA,RI)=Z:NEXT RI:NEXT RA
1000 GRAPHICS 8:SETCOLOR 2,1,0:GOTO 10
40
1021 T$="N, ":RETURN
1022 T$="S, ":RETURN
1023 T$="O, ":RETURN
```



```
1024 T$="E, ":RETURN
1025 T$="AR, ":RETURN
1026 T$="AB, ":RETURN
1040 T$=""
1050 ZE$=CHR$(30):ZE$(LEN(ZE$)+1)=CHR$
(30):ZE$(LEN(ZE$)+1)="."
1080 PRINT :ZUG=ZUG+1:IF ALT<>SP THEN
GRAPHICS 8:SETCOLOR 2,1,0:GOSUB 30000
1100 POSITION 2,0:PRINT "ESTOY ";:GOSU
B 200+SP:ALT=SP
1110 DZ$="VEO: "
1120 FOR I=1 TO AO
1130 IF OB(I)<>SP THEN GOTO 1170
1140 GD=-1:GOSUB 300+I:T$(LEN(T$)+1)="
, "
1150 IF LEN(DZ$)+LEN(T$)>=38 THEN PRIN
T DZ$:DZ$=T$(1,LEN(T$)):T$="":GOTO 117
0
1160 DZ$(LEN(DZ$)+1)=T$:T$=""
1170 NEXT I
1180 IF  NOT GD THEN DZ$(LEN(DZ$)+1)="
NADA ESPECIAL. "
1190 DZ$(LEN(DZ$)+1)=ZE$
1200 PRINT DZ$:GD=0:DZ$=""
1230 FOR RI=1 TO 6
1240 IF DU(SP,RI)=0 THEN GOTO 1280
1250 GOSUB 1020+RI:GD=-1
1270 DZ$(LEN(DZ$)+1)=T$:T$=""
1280 NEXT RI
1290 IF  NOT GD THEN DZ$(LEN(DZ$)+1)="
NINGUN SITIO"
1300 PRINT DZ$;
1350 IF FL(1)=0 THEN PRINT "ALARMA!";:
IF ZUG>9 THEN MO$="NOS HA ABORDADO UN
ASTEROIDE.":GOTO 4500
1390 INPUT EINGABE$
1400 IF EINGABE$="N" AND DU(SP,1)<>0 T
HEN SP=DU(SP,1):PRINT OK$:GOTO 1080
1410 IF EINGABE$="S" AND DU(SP,2)<>0 T
HEN SP=DU(SP,2):PRINT OK$:GOTO 1080
```

```
1420 IF EINGABE$="O" AND DU(SP,3)<>0 T
HEN SP=DU(SP,3):PRINT OK$:GOTO 1080
1430 IF EINGABE$="E" AND DU(SP,4)<>0 T
HEN SP=DU(SP,4):PRINT OK$:GOTO 1080
1440 IF EINGABE$="AR" AND DU(SP,5)<>0
THEN SP=DU(SP,5):PRINT OK$:GOTO 1080
1450 IF EINGABE$="AB" AND DU(SP,6)<>0
THEN SP=DU(SP,6):PRINT OK$:GOTO 1080
1460 IF LEN(EINGABE$)<3 THEN PRINT "HA
CIA ALLI NO CONDUCE NINGUN CAMINO!":GO
TO 1080
1470 IF LEN(EINGABE$)>8 THEN GOTO 2000
1500 IF EINGABE$(1,3)<>"INV" THEN GOTO
 1600
1505 PRINT "INVENTARIO:"
1510 FOR I=1 TO AO
1515 IF OB(I)=-1 THEN GOSUB 300+I:PRIN
T T$;" ";
1520 NEXT I
1525 GOTO 1080
1600 IF EINGABE$(1,3)<>"SAV" THEN GOTO
 1700
1605 PRINT "NOMBRE ";:INPUT EINGABE$:I
F LEN(EINGABE$)>8 THEN GOTO 1605
1610 T$="D:":EINGABE$(LEN(EINGABE$)+1)
=".DAT"
1615 T$(LEN(T$)+1)=EINGABE$
1620 OPEN #1,8,0,T$
1625 PRINT #1,SP
1630 FOR I=1 TO AO:PRINT #1,OB(I):NEXT
 I
1635 FOR RA=1 TO AR:FOR RI=1 TO 6:PRIN
T #1,DU(RA,RI):NEXT RI:NEXT RA
1655 FOR I=1 TO AF:PRINT #1,FL(I):NEXT
 I
1660 CLOSE #1:PRINT OK$:GOTO 1080
1700 IF EINGABE$(1,3)<>"LOA" THEN GOTO
 1900
1705 PRINT "QUE JUEGO ";:INPUT EINGABE
$:IF LEN(EINGABE$)>8 THEN 1705
```



```
1710 T$="D:":EINGABE$(LEN(EINGABE$)+1)
=".DAT"
1715 T$(LEN(T$)+1)=EINGABE$
1720 OPEN #1,4,0,T$
1725 INPUT #1,SP
1730 FOR I=1 TO AO:INPUT #1,LO:OB(I)=L
O:NEXT I
1735 FOR RA=1 TO AR
1740 FOR RI=1 TO 6
1745 INPUT #1,ZIEL:DU(RA,RI)=ZIEL
1750 NEXT RI
1755 NEXT RA
1760 FOR I=1 TO AF:INPUT #1,ZIEL:FL(I)
=ZIEL:NEXT I
1765 CLOSE #1:PRINT OK$:GOTO 1080
1900 IF EINGABE$(1,3)<>"INS" THEN GOTO
 1910
1910 IF EINGABE$(1,3)<>"FIN" THEN PRIN
T "NO COMPRENDO":GOTO 1080
1915 GRAPHICS 0:PRINT "EL AUTOR LE DES
EA MAS SUERTE EN EL":PRINT "PROXIMO IN
TENTO!":PRINT :PRINT :PRINT :END
2000 LN=LEN(EINGABE$)
2010 FOR I=1 TO LN
2020 IF EINGABE$(I,I)<>" " THEN NEXT I
2030 EV$=EINGABE$(1,I)
2040 IF LEN(EV$)=LN-1 THEN GOTO 2090
2050 EO$=EINGABE$(I+1,LN)
2060 VN=0:N=0
2070 EV$=EV$(1,4)
2080 EO$=EO$(1,4)
2090 FOR I=1 TO LEN(VERBEN$) STEP 10
2100 VN=VN+1
2110 IF VERBEN$(I,I+3)=EV$ THEN GOTO 2
140
2120 NEXT I
2130 PRINT "NO ENTIENDO EL VERBO!":GOT
O 1080
2140 FOR I=1 TO LEN(OBJEKTE$)-2 STEP 4
2150 N=N+1
```

```
2160 IF OBJEKTE$(I,I+3)=EO$ THEN 2200
2170 NEXT I
2180 PRINT "NO ENTIENDO EL OBJETO!":GO
TO 1080
2200 ON VN GOTO 5000,6000,7000,8000,90
00,10000,11000,12000,13000,14000
2330 RETURN
4500 GRAPHICS 0
4510 PRINT "":PRINT :PRINT MO$
4520 PRINT :PRINT "ESTOY MUERTO!":FOR
I=1 TO 800:NEXT I:GOTO 1915
4530 PRINT"LO VUELVE A INTENTAR";:INP
UT ENTRADA$
4540 IF ENTRADA$(1,1)="S"THENRUN
4550 END
4800 GRAPHICS 0
4810 PRINT "ENHORABUENA!"
4820 PRINT :PRINT :PRINT "HA RESUELTO
EL PROBLEMA PLANTEADO Y"
4830 PRINT :PRINT "PUEDE AFRONTARSE A
UNA NUEVA "
4840 PRINT :PRINT "AVENTURA."
4850 PRINT :PRINT :PRINT :END
5000 IF OB(N)<>SP AND OB(N)<>-1 THEN G
OTO 5900
5005 IF O=1 THEN PRINT "CON ESTO CONTR
OLO EL PILOTO AUTOMATICO.":PRINT "VEO
TECLAS:*SD*,SELO,STAT,VIEW.":OB(22)=1
5006 IF N=2 AND FL(1)=0 THEN PRINT "LA
 PANTALLA MUESTRA UN ASTEROIDE":GOTO 1
080
5007 IF N=2 AND FL(1) THEN PRINT "NO V
EO NADA ESPECIAL.":GOTO 1080
5010 IF N=6 THEN GOSUB 601:GOTO 1080
5013 IF N=25 AND OB(40)=0 THEN PRINT "
DEBAJO HABIA UNA TARJETA MAGNETICA.":O
B(40)=11:GOSUB 30000:GOTO 1080
5014 IF N=8 AND SP=23 AND FL(7)=0 THEN
 PRINT "ESTA CERRADO.":GOTO 1080
5015 IF N=8 AND SP=25 AND FL(8)=0 THEN
 PRINT "ESTA CERRADO.":GOTO 1080
5016 IF N=8 AND SP=23 AND FL(7)=-1 THE
N PRINT "CONDUCE HACIA UN ENTREPUENTE.
```



```
":GOTO 1080
5017 IF N=8 AND SP=25 AND FL(8)=-1 THE
N PRINT "CONDUCE HACIA UN ENTREPUENTE.
":GOTO 1080
5019 IF N=39 THEN GOSUB 601:OB(5)=SP:O
B(6)=SP:OB(39)=0:GOTO 1080
5020 IF N=5 THEN GOTO 20000
5021 IF N=44 AND OB(29)=0 THEN PRINT "
DEBAJO ESTA MI TRAJE ESPACIAL.":GOTO 1
080
5025 IF N=4 THEN PRINT "EXISTEN LAS CI
FRAS 0 - 9.":GOTO 1080
5026 IF N=10 AND OB(32)<>-1 THEN PRINT
 "NO PUEDO ABRIRLO.":GOTO 1080
5027 IF N=10 AND OB(32)=-1 THEN PRINT
"DENTRO ESTA EL ORDENADOR CENTRAL.":OB
(11)=7:OB(10)=0:GOTO 1080
5028 IF N=11 THEN PRINT "VEO NUMEROSOS
 MODULOS.":OB(13)=SP:OB(15)=0:GOTO 108
0
5029 IF N=13 THEN PRINT "SON MODULOS D
E EXPANSION.":GOTO 1080
5030 IF N=12 THEN PRINT "... UN MODULO
 NORMAL DE ORDENADOR.":GOTO 1080
5031 IF N=14 THEN GOSUB 601:GOTO 1080
5032 IF N=15 THEN GOSUB 601:GOTO 1080
5033 IF N=16 THEN PRINT "PERTENECEN AL
 SISTEMA DE VENTILACION.":GOTO 1080
5034 IF N=17 THEN PRINT "SOLO UN INDIC
ADOR.":GOTO 1080
5035 IF N=20 THEN GOSUB 601:GOTO 1080
5036 IF N=33 THEN GOSUB 601:OB(30)=SP:
GOTO 1080
5037 IF N=22 THEN PRINT "VEO TECLAS: L
AND MOVE STOP SETC":OB(N)=0:FOR I=1 TO
 2000:NEXT I:GOTO 1080
5038 IF N=23 THEN GOSUB 601:GOTO 1080
5039 IF N=24 AND OB(32)=0 THEN PRINT "
DETRAS HAY UN ATORNILLADOR.":GOTO 1080
5040 IF N=32 THEN GOSUB 601:GOTO 1080
```

```
5041 IF N=27 THEN GOSUB 601:GOTO 1080
5042 IF N=28 AND FL(4)=0 THEN PRINT "E
STA VACIA.":GOTO 1080
5043 IF N=28 AND FL(4)=-1 THEN PRINT "
ESTA LLENA.":GOTO 1080
5044 IF N=29 THEN PRINT "ES EL MIO.":G
OTO 1080
5045 IF N=30 THEN GOSUB 601:GOTO 1080
5046 IF N=48 THEN GOSUB 601:GOTO 1080
5200 IF N=24 THEN GOSUB 601:GOTO 1080
5899 GOSUB 601:GOTO 1080
5900 REM
5901 IF N=7 AND SP=24 THEN GOSUB 601:G
OTO 1080
5902 IF N=7 AND SP=23 THEN FL(5)=-1:AL
T=0:OB(8)=SP:GOTO 1080
5903 IF N=7 AND SP=24 THEN GOSUB 601:G
OTO 1080
5904 IF N=7 AND SP=25 THEN FL(6)=-1:OB
(9)=SP:ALT=1:GOTO 1080
5905 IF N=7 AND SP=26 THEN GOSUB 601:G
OTO 1080
5906 IF N=7 AND SP=27 THEN GOSUB 601:G
OTO 1080
5907 IF N=7 AND SP=22 THEN GOSUB 601:G
OTO 1080
5910 IF N=20 AND SP=8 THEN ? "PIA III
ESTA INDICADO CON 151064.":GOTO 1080
5916 IF N=25 AND OB(40)=-1 THEN GOTO 1
080
5920 IF N=32 THEN GOSUB 601:GOTO 1080
5922 IF N=8 AND SP=25 AND DU(25,4)=0 T
HEN PRINT "ESTA CERRADO.":GOTO 1080
5923 IF N=47 AND SP=2 THEN PRINT "EL M
ISMO MODELO!":GOTO 1080
5924 IF N=4 AND SP=2 THEN PRINT "EL MI
SMO MODELO!":GOTO 1080
5925 IF N=20 AND SP=15 THEN PRINT "SON
 LOS MEDICAMENTOS.":GOTO 1080
5990 PRINT "AQUI NO VEO NADA PARECIDO!
```



```
":GOTO 1080
6000 IF OB(N)<>SP AND OB(N)<>-1 THEN G
OTO 6900
6005 IF N=40 THEN PRINT OK$:OB(N)=-1:G
OSUB 30000:GOTO 1080
6010 IF N=12 AND OB(12)=-1 THEN GOSUB
604:GOTO 1080
6011 IF N=12 THEN OB(12)=-1:PRINT OK$:
GOTO 1080
6012 IF N=13 THEN PRINT "TENGO UN MODU
LO.":OB(36)=-1:GOTO 1080
6013 IF N=14 OR N=28 THEN PRINT OK$:OB
(N)=-1:GOTO 1080
6014 IF (N=30 OR N=50) THEN PRINT OK$:
OB(N)=-1:GOTO 1080
6800 IF N=32 THEN PRINT OK$:OB(N)=-1:G
OTO 1080
6890 IF N=SP THEN GOSUB 602:GOTO 1080
6900 REM
6910 IF N=29 THEN PRINT OK$:OB(N)=-1:G
OTO 1080
6915 IF N=20 AND SP=15 THEN PRINT OK$:
OB(50)=-1:GOTO 1080
6920 IF N=32 AND OB(N)=0 THEN OB(32)=-
1:PRINT OK$:GOTO 1080
6990 PRINT "AQUI NO VEO NADA PARECIDO.
":GOTO 1080
7000 IF N=41 AND OB(40)=-1 THEN PRINT
OK$:DU(16,4)=1:ALT=2:GOTO 1080
7005 IF N=41 AND OB(40)<>-1 THEN PRINT
 "NO ES POSIBLE EN ESTE MOMENTO.":GOTO
 1080
7010 IF N=8 AND OB(14)<>-1 THEN PRINT
"NO TENGO NADA PARA ABRIR.":GOTO 1080
7011 IF N=8 AND SP=23 AND OB(14)=-1 TH
EN PRINT OK$:DU(23,4)=7:FL(7)=-1:GOTO
1080
7015 IF N=23 AND SP=9 THEN MO$="OXIGEN
O SE ESCAPA Y ME ARREBATA AL    ESPACI
O.":GOTO 4500
```

```
7020 IF N=9 AND SP=14 THEN MO$="ME ALC
ANZO UNA DESCARGA RADIADA DE LA SALA D
E REACTORES.":GOTO 4500
7099 GOSUB 605:GOTO 1080
8000 REM
8010 IF N=46 AND FL(2)=-1 THEN PRINT "
CLIC":DU(24,6)=25:GOTO 1080
8012 IF N=46 AND FL(2)=0 THEN PRINT "N
O SUCEDE NADA.":GOTO 1080
8020 IF N=45 THEN GOTO 20200
8021 IF N=43 THEN GOTO 20300
8022 IF N=42 THEN MO$="ESTO FUE LA AUT
ODESTRUCCION!":GOTO 4500
8024 IF N=18 THEN MO$="EL PILOTO AUTOM
ATICO ATERRIZO LA NAVE EN EL SOL.":GOT
O 4500
8025 IF N=26 AND TM$="151063" THEN MO$
="LA FUNCION DE AVERIA! COORDENADAS
  ERRONEAS.":GOTO 4500
8026 IF N=26 AND FL(1)=0 THEN PRINT "L
A NAVE ESQUIVA EL ASTEROIDE.";:FL(1)=-
1:GOTO 1080
8027 IF N=26 AND FL(1)=-1 AND FL(10)=0
 THEN MO$="DATOS DE RUMBO NO INDICADOS
 - DESTINO FUE UN SOL.":GOTO 4500
8028 IF N=26 AND FL(10)=-1 AND OB(50)<
>13 THEN MO$="LINCHADO, POR HABER LLEG
ADO SIN MEDI- CAMENTOS.":GOTO 4500
8029 IF N=31 THEN FL(1)=0:PRINT OK$:GO
TO 1080
8030 IF N=35 THEN PRINT "COORDENADAS "
;:INPUT TM$:IF TM$="151064" THEN FL(10
)=-1:GOTO 1080
8031 IF N=35 THEN GOTO 1080
8032 IF N=26 AND FL(10)=-1 AND OB(50)=
13 THEN GOTO 4800
8115 IF N=23 AND SP=9 THEN MO$="SE ESC
APA OXIGENO Y ME ARREBATA AL    ESPACI
O.":GOTO 4500
8990 PRINT "AQUI NO VEO NINGUN BOTON C
```



```
OMO ESTE.":GOTO 1080
9000 IF OB(N)<>-1 THEN GOTO 9900
9010 PRINT OK$:OB(N)=SP:GOTO 1080
9900 IF N=13 AND OB(36)=-1 THEN OB(36)
=SP:PRINT OK$:GOTO 1080
9915 IF N=20 AND OB(50)=-1 AND  NOT (S
P=15 OR SP=13) THEN PRINT OK$:OB(50)=S
P:FL(11)=-1:GOTO 1080
10000 IF N=29 THEN PRINT "ME HE PUESTO
 EL TRAJE ESPACIAL.":OB(29)=-2:OB(14)=
-1:GOTO 1080
10010 GOTO 6000
11000 IF N=28 AND SP<>21 THEN PRINT "C
ON QUE?":GOTO 1080
11010 IF N=28 AND SP=21 AND OB(30)<>-1
 THEN PRINT "NO ES POSIBLE EN ESTE MOM
ENTO.":GOTO 1080
11020 IF N=28 AND SP=21 AND OB(30)=-1
THEN PRINT OK$:FL(4)=-1:GOTO 1080
11900 GOSUB 605:GOTO 1080
12000 IF N=38 AND SP=12 THEN INPUT TM$
:GOTO 1080
12005 IF N=38 AND SP=2 THEN INPUT TM$:
GOTO 1080
12010 IF N=49 AND SP=6 THEN INPUT TM$:
IF TM$="220559" THEN DU(2,1)=15:SP=15:
GOTO 1080
12020 IF N=49 AND SP=6 AND TM$<>"22055
9" THEN MO$="SE HAN ACTIVADO LOS SISTE
MAS DE DE-    FENSA.":GOTO 4500
13000 IF N=47 AND  NOT (SP=12 OR SP=2)
 THEN PRINT "NO VEO NINGUN TRANSMISOR.
":GOTO 1080
13010 IF N=47 AND SP=12 AND TM$<>"6446
64" THEN MO$="MIS ATOMOS FLOTAN EN EL
HIPERESPACIO.":GOTO 4500
13011 IF N=47 AND SP=2 AND TM$<>"64466
3" THEN MO$="MIS ATOMOS FLOTAN EN EL H
IPERESPACIO.":GOTO 4500
13020 IF N=47 AND (SP=12 OR SP=2) AND
```

```
TM$="644664" THEN SP=2:GOTO 1080
13021 IF N=47 AND (SP=12 OR SP=2) AND
TM$="644663" THEN SP=12:GOTO 1080
13030 IF N=22 AND SP=1 THEN FL(1)=-1:P
RINT OK$:GOTO 1080
13900 GOSUB 605:GOTO 1080
14000 IF N=13 AND SP=7 AND OB(13)=7 TH
EN PRINT OK$:OB(12)=0:OB(36)=-1:FL(2)=
-1:GOTO 1080
20000 GRAPHICS 0:? "ACCIDENTE EN PIA 3
 * REQUERIDOS MEDI- CAMENTOS CIBARAD 9
2 * VOLAR CON RUMBO A ESTACION 1701 *"
20010 ? :? "COORDENADAS 1701: 644664":
? "PIA3: 151063":? :? "PASSWORD: 22055
9";? :? "*FUNCION DE AVERIA*":ALT=0:GO
TO 1390
20200 IF FL(1)=0 THEN ? "PILOTO AUTOMA
TICO OFF"
20210 IF FL(2)=0 THEN ? "CPU DEFECTUOS
O"
20220 IF FL(2)=-1 THEN ? "SISTEMAS FUN
CIONANDO"
20230 IF FL(1)=0 THEN ? "ADVERTENCIA D
E COLISION"
20240 IF FL(2)=-1 THEN ? "SITUACION: 6
44663"
20250 GOTO 1390
21100 GOSUB 31000:GOSUB 31600:RETURN
21300 PLOT 100,80:DRAWTO 200,80:DRAWTO
 200,120:DRAWTO 100,120:DRAWTO 100,80:
DRAWTO 110,70:DRAWTO 190,70:DRAWTO 200
,80
21308 PLOT 180,70:DRAWTO 180,60:PLOT 1
20,60:DRAWTO 120,70:PLOT 110,60:DRAWTO
 190,60:DRAWTO 190,0:DRAWTO 110,0:DRAW
TO 110,60
21316 PLOT 100,100:DRAWTO 0,100:PLOT 2
00,100:DRAWTO 230,100:DRAWTO 270,140:D
RAWTO 270,20:DRAWTO 230,0
21320 PLOT 150,70:DRAWTO 150,80:RETURN
```



```
21700 PLOT 99,107:PLOT 99,132:PLOT 164
,132:PLOT 174,132:PLOT 244,132:PLOT 24
4,107:PLOT 174,107
21708 PLOT 0,0:DRAWTO 30,10:DRAWTO 30,
80:DRAWTO 0,100
21711 PLOT 90,140:DRAWTO 90,100:DRAWTO
 250,100:DRAWTO 250,140:DRAWTO 90,140
21715 PLOT 90,100:DRAWTO 110,80:DRAWTO
 230,80:DRAWTO 250,100
21718 PLOT 250,120:DRAWTO 310,120
21719 PLOT 90,120:DRAWTO 50,120:DRAWTO
 10,160:PLOT 50,120:DRAWTO 50,0:PLOT 1
30,80:DRAWTO 130,60
21723 PLOT 131,60:DRAWTO 131,79:PLOT 1
23,87:DRAWTO 119,91:DRAWTO 122,91:DRAW
TO 126,87:DRAWTO 124,87:RETURN
22000 PLOT 43,60:PLOT 43,59:PLOT 10,10
:DRAWTO 40,10:DRAWTO 50,30:DRAWTO 50,9
0:DRAWTO 10,90:PLOT 50,30:DRAWTO 10,30
22007 PLOT 0,149:DRAWTO 240,149:DRAWTO
 260,160:PLOT 239,149:DRAWTO 239,0
22010 PLOT 219,20:DRAWTO 199,20:DRAWTO
 199,0:DRAWTO 179,20:DRAWTO 159,20:DRA
WTO 159,40:DRAWTO 179,40:DRAWTO 179,60
22011 PLOT 199,20:DRAWTO 199,0:DRAWTO
179,0:DRAWTO 179,20
22018 PLOT 179,60:DRAWTO 199,60:DRAWTO
 199,40:DRAWTO 219,40:DRAWTO 219,20
22025 PLOT 79,140:DRAWTO 219,140:DRAWT
O 199,120:DRAWTO 59,120:DRAWTO 79,140:
RETURN
22100 GRAPHICS 8:SETCOLOR 2,1,0
22101 PLOT 20,0:DRAWTO 20,100
22102 DRAWTO 0,120
22103 PLOT 100,82:DRAWTO 180,80
22104 DRAWTO 210,50:DRAWTO 130,50:DRAW
TO 100,80
22107 PLOT 110,80:DRAWTO 110,110:PLOT
170,110:DRAWTO 170,80:PLOT 200,80:DRAW
```

```
TO 200,60:PLOT 20,60:DRAWTO 120,60
22108 PLOT 200,60
22110 DRAWTO 260,60:DRAWTO 280,80:DRAW
TO 280,0:DRAWTO 310,30:DRAWTO 310,110:
DRAWTO 314,114:PLOT 260,60:DRAWTO 260,
0
22118 PLOT 20,60
22120 IF OB(40)<>11 THEN GOTO 22130
22122 PLOT 130,70:DRAWTO 140,70:DRAWTO
 135,75:DRAWTO 125,75:DRAWTO 130,70
22130 RETURN
22200 PLOT 210,0
22201 PLOT 218,157
22202 PLOT 215,157
22203 PLOT 212,157
22204 PLOT 209,157
22205 PLOT 206,157
22206 PLOT 203,157
22207 PLOT 200,157
22208 PLOT 197,157
22209 PLOT 194,157
22210 PLOT 191,157
22211 PLOT 210,0:DRAWTO 250,30
22212 PLOT 250,110:DRAWTO 250,30
22213 PLOT 210,0:DRAWTO 170,30
22214 PLOT 170,30:DRAWTO 170,110
22215 PLOT 170,110:DRAWTO 250,110
22216 PLOT 240,160:DRAWTO 220,140
22217 PLOT 220,140:DRAWTO 120,140
22218 PLOT 120,140:DRAWTO 100,160
22219 PLOT 110,160:DRAWTO 120,150
22220 PLOT 120,150:DRAWTO 220,150
22221 PLOT 220,150:DRAWTO 230,160
22222 PLOT 61,0:DRAWTO 61,70
22223 PLOT 61,70:DRAWTO 111,20
22224 PLOT 111,20:DRAWTO 111,0
22225 PLOT 111,20:DRAWTO 181,20
22226 PLOT 238,20:DRAWTO 318,20
22227 PLOT 220,80:DRAWTO 250,110
22228 PLOT 170,110:DRAWTO 200,80
```



```
22229 PLOT 200,80:DRAWTO 220,80
22230 PLOT 250,30:DRAWTO 220,60
22231 PLOT 220,60:DRAWTO 200,60
22232 PLOT 200,60:DRAWTO 170,30
22233 PLOT 30,10:DRAWTO 30,100
22234 PLOT 30,100:DRAWTO 10,120:RETURN

22300 PLOT 0,0
22301 PLOT 0,0:DRAWTO 40,40
22302 PLOT 40,40:DRAWTO 160,40:DRAWTO
300,0
22304 PLOT 280,20:DRAWTO 20,20
22305 PLOT 40,40:DRAWTO 40,100:DRAWTO
260,100:DRAWTO 260,40
22308 PLOT 260,100:DRAWTO 300,140
22309 PLOT 40,100:DRAWTO 30,110:DRAWTO
 30,50:DRAWTO 0,20:DRAWTO 0,140
22315 IF (OB(50)=13 OR SP=15) THEN GOS
UB 22122
22316 RETURN
22400 PLOT 60,0:DRAWTO 60,100
22402 PLOT 60,100:DRAWTO 80,80
22403 PLOT 80,40:DRAWTO 80,120
22404 PLOT 80,120:DRAWTO 160,120
22405 PLOT 160,120:DRAWTO 160,80
22406 PLOT 160,80:DRAWTO 140,80
22407 PLOT 140,80:DRAWTO 100,40
22408 PLOT 100,40:DRAWTO 80,40
22409 PLOT 80,40:DRAWTO 120,0
22410 PLOT 100,40:DRAWTO 140,0
22411 PLOT 140,80:DRAWTO 220,0
22412 PLOT 240,0:DRAWTO 160,80
22413 PLOT 160,120:DRAWTO 260,20
22414 PLOT 260,20:DRAWTO 260,0
22415 PLOT 260,10:DRAWTO 280,10
22416 PLOT 280,10:DRAWTO 280,0:RETURN
22600 PLOT 100,0:DRAWTO 200,0:DRAWTO 2
00,140:DRAWTO 100,140:DRAWTO 100,0
22602 IF SP=9 THEN PLOT 150,0:DRAWTO 1
50,140
```

```
22605 IF DU(16,4)<>1 AND SP=9 THEN PLO
T 150,0:DRAWTO 150,140
22606 PLOT 220,60:DRAWTO 220,80:DRAWTO
 210,80:DRAWTO 210,60:DRAWTO 220,60
22610 PLOT 50,142:DRAWTO 260,142:DRAWT
O 260,0
22612 PLOT 50,142:DRAWTO 50,0
22613 PLOT 50,142:DRAWTO 32,160
22614 PLOT 260,142:DRAWTO 278,160
22630 IF SP=9 OR SP=6 THEN GOTO 22699
22631 IF DU(16,4)<>1 THEN GOTO 22699
22643 PLOT 170,0:DRAWTO 170,60:DRAWTO
200,30
22645 PLOT 170,60:DRAWTO 160,60:DRAWTO
 150,70:DRAWTO 150,50:DRAWTO 160,40:DR
AWTO 160,60
22650 PLOT 150,70:DRAWTO 100,70
22651 PLOT 100,50:DRAWTO 150,50
22652 PLOT 160,40:DRAWTO 100,40
22653 PLOT 100,30:DRAWTO 160,30
22654 PLOT 160,30:DRAWTO 160,0
22699 RETURN
22700 PLOT 10,0:DRAWTO 10,100:DRAWTO 0
,110:PLOT 130,50:DRAWTO 10,50:PLOT 0,1
20:DRAWTO 310,120
22704 PLOT 310,90:DRAWTO 250,90:DRAWTO
 250,0
22706 PLOT 250,50:DRAWTO 170,50:DRAWTO
 170,0:PLOT 170,10:DRAWTO 130,50:DRAWT
O 130,0:RETURN
23000 ? "AQUI ESTA DEMASIADO OSCURO PA
RA VER.":PRINT "E.";:INPUT EINGABE$:IF
 EINGABE$(1,4)<>"TECL" THEN 23090
23010 IF EINGABE$="E" THEN SP=12
23020 PRINT "AQUI HAY UNA BOTELLA DE O
XIGENO.":OB(28)=20
23090 RETURN
23200 GOSUB 31200:GOSUB 31400:GOSUB 31
300:RETURN
23300 GOSUB 31200:IF FL(5) AND SP=23 T
```



```
HEN GOSUB 31400
23310 RETURN
23400 GOSUB 31200:GOSUB 31300:GOSUB 31
400:RETURN
23600 GOSUB 31200:GOSUB 31400:RETURN
23700 GOSUB 31200:GOSUB 31400:RETURN
24000 PLOT 3,103:DRAWTO 73,103:DRAWTO
103,73:DRAWTO 103,53:DRAWTO 73,83:DRAW
TO 73,103
24006 PLOT 73,93:DRAWTO 3,93
24007 PLOT 93,63:DRAWTO 3,63
24008 PLOT 103,72:DRAWTO 223,72
24009 PLOT 223,0:DRAWTO 223,72:DRAWTO
263,112:DRAWTO 263,12:DRAWTO 253,2
24013 PLOT 263,62:DRAWTO 303,62
24014 PLOT 313,160:DRAWTO 303,150:DRAW
TO 303,50:DRAWTO 253,0:RETURN
30000 IF SP>20 THEN 30030
30005 IF SP>10 THEN 30020
30010 ON SP GOSUB 21300,22200,21300,21
400,21500,22600,21700,21300,22600,2200
0
30011 RETURN
30020 ON SP-10 GOSUB 22100,22200,22300
,22400,22300,22600,22700,22700,22700,2
3000
30021 RETURN
30030 ON SP-20 GOSUB 23100,23200,23300
,23400,23300,23600,23700,23800,23900,2
4000
30031 RETURN
31200 PLOT 80,0:DRAWTO 80,160
31202 PLOT 120,140:DRAWTO 140,140:PLOT
 200,140:DRAWTO 220,140:PLOT 260,160:D
RAWTO 260,0
31203 PLOT 200,140:DRAWTO 220,140
31204 PLOT 260,160:DRAWTO 260,0
31205 PLOT 220,100:DRAWTO 200,100:PLOT
 200,60:DRAWTO 220,60:PLOT 220,20:DRAW
TO 200,20
```

```
31208 PLOT 140,20:DRAWTO 120,20:PLOT 1
20,60:DRAWTO 140,60
31210 PLOT 140,100:DRAWTO 120,100:RETU
RN
31300 PLOT 60,80
31301 PLOT 20,20:DRAWTO 60,40:DRAWTO 6
0,100:DRAWTO 20,140:RETURN
31400 PLOT 280,40:DRAWTO 300,20
31402 PLOT 280,40:DRAWTO 280,100:DRAWT
O 300,120:IF FL(5) AND SP=23 THEN PLOT
 285,70
31405 RETURN
```


PROGRAMADOR

DE

GRAFICOS

Ya he desarrollado el programa, cuyo listado mostramos a continuación, para poder programar sencillas gráficas HIRES de forma fácil y rápida.

El procedimiento seguido es muy sencillo: usted crea la gráfica en la pantalla, con la ayuda de algunas preguntas hechas por el editor,y utilizando un cursor controlado por teclas. Se graban los datos necesarios para crear la imagen y se utilizan, más adelante, para confeccionar un subprograma, que se podrá añadir a otros programas.

De esta forma evitamos tener que emplear algún medio auxiliar, como por ejemplo, hojas de trabajo de la pantalla, para determinar las coordenadas apropiadas.

CONFECCION DE GRAFICAS CON GRAFPO

Antes de empezar, compruebe si su Atari se encuentra en el modo gráfico mayusculas.

Después de mostrar la imagen titular, el editor pedirá el modo operativo deseado. Esto quiere decir que usted seleccionará el modo gráfico y los colores del fondo y del primer plano.

Pulse una tecla cualquiera en cada caso; si el color responde a sus deseos, pulse <RETURN>.

Cuando el ordenador haya cambiado al modo operativo deseado, podrá disponer de un conjunto de instrucciones de control y de carácteres.

Las instrucciones de carácteres se ejecutan al pulsar su tecla correpondiente, mientras que debe pulsarse la tecla <CONTROL> para procesar las instrucciones de control.

MOVIMIENTOS DEL CURSOR

El cursor es un punto intermitente e indestructible que aparece en la pantalla. Su control se efectúa con las teclas:

```
Q W E
A   D
Z X C
```

Su disposición en el teclado corresponde a las direcciones del movimiento.

Al iniciar el programa, el cursor se encuentra siempre en la posición 0,0, o sea, en la esquina superior izquierda.

Para que podamos dirigirnos rápidamente a cualquier punto en la pantalla, tenemos la posibilidad de determinar cualquier longitud de paso del cursor después de entrar CTRL-C.

La alternativa prevé una pulsación de la tecla P, y después podrá entrar directamente las nuevas coordenadas.

DIBUJAR

Para dibujar un punto, dirija el cursor a la posición deseada y pulse U para su recepción.

El punto será posicionado inmediatamente, y se memorizan sus coordenadas para utilizarlos en la posterior programación.

Si desea dibujar una línea, dirija el cursor primero al punto de partida.

Después pulse la tecla L para fijar este punto, dirija el cursor ahora al punto final y vuelva a pulsar L.

TRATAMIENTO DE ERRORES

Para evitar que se reciban valores incorrectos, es necesario confirmar la posición de cada punto y de cada línea dibujada con J. Al pulsar cualquier otra tecla,se anulará el último proceso realizado.

Por razones de rapidez, es mejor prescindir de esta confirmación. Con CTRL-B es posible activar y desactivar esta función de control en cualquier momento.

Aunque incurra en un error, su obra no estará perdida. Al entrar CTRL-Z se borra la pantalla y se vuelve a dibujar la imagen, pero sin tener en cuenta ni la última línea dibujada ni el último punto.

GRABAR IMAGEN, CARGAR IMAGEN

Pulse CTRL-L o CTRL-S y entre un nombre de un máximo de ocho letras para la imagen. Grafpo añade automáticamente el complemento .BDT (datos de imagen) y ejecuta el proceso deseado.

PROGRAMAR IMAGEN

Después de pulsar simultaneamente CONTROL y P, se deberá entrar otra vez un nombre. A continuación, se le pedirá el número de línea en el cual debe comenzar el programa, y se crea un correspondiente programa.

CONTROL Q finaliza la ejecución del programa.

EMPLEO DE LOS PROGRAMAS CREADOS

Los programas generados no responden al formato habitual con el que se graban programas Basic. Por esta razón, no podemos cargarlos con la instrucción LOAD sino con ENTER.

Tenga en cuenta, que al cargar un programa con ENTER no se borra el actual contenido de la memoria. Y puede ser útil para enlazar diversas partes del programa, pero también puede ocurrir que los listados se mezclen de manera confusa.


```
1 REM PROGRAMADOR GRAFICO
2 REM (C) 1984 BY JOERG WALKOWIAK
3 REM RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
5 GRAPHICS 0:SETCOLOR 2,0,0
6 PRINT "     EDITOR & PROGRAMADOR GRAF
ICO              (C) 1984 BY DATA BECKER
        ":PRINT :PRINT :PRINT
10 SETCOLOR 2,0,0:GOTO 100
19 REM ******************** SUBPROGRAMAS
20 PRINT #1,LINEA,Z$:LINEA=LINEA+10:RE
TURN
40 Z$(LEN(Z$)+1)=H$:RETURN
80 Z$(LEN(Z$)+1)=STR$(X):RETURN
99 REM ***************** INICIALIZACION
100 CLR :DIM PX(200),PY(200),PC(200),V
X(100),VY(100),NX(100),NY(100),NOMBRE$
(8),FILE$(12),Z$(50),H$(40)
110 PZ=0:LZ=0:C=1:PX(0)=0:PY(0)=0:PX(1
)=0:PY(1)=0
120 XMAX=320:YMAX=160
130 REVISION=0
200 OPEN #2,4,0,"K:"
205 HF=0:PRINT "POR FAVOR, ELIJA ...":
PRINT :PRINT
207 POSITION 2,23:PRINT "SI CORRECTO,
PULSAR <RETURN> ...";
210 POSITION 2,9:PRINT "... COLOR DE F
ONDO":GET #2,E:IF E=155 THEN GOTO 230
215 HF=HF+1:IF HF=16 THEN HF=0
220 PRINT HF:SETCOLOR 2,HF,0:GOTO 210
225 HE=0
230 POSITION 2,11:PRINT "... CLARIDAD"
:GET #2,E:IF E=155 THEN GOTO 250
235 HE=HE+1:IF HE=15 THEN HE=0
240 PRINT HE:SETCOLOR 2,HF,HE:GOTO 230
245 GET #2,E:IF E<>155 THEN GOTO 205
250 POSITION 2,9:PRINT "
       ":POSITION 5,9:PRINT "
     "
260 POSITION 2,9:PRINT "... MODO.OPERA
```

```
TIVO GRAFICO 8";CHR$(30);CHR$(30);:INP
UT MODUS
270 POSITION 2,11:PRINT "... COLOR DE
CARACTER 1";CHR$(30);CHR$(30);:INPUT S
F
280 GRAPHICS MODUS
290 SETCOLOR 2,HF,HE:COLOR SF
300 TRAP 10000:REM INTERCEPTAR ERRORES
999 REM ********** PROGRAMA PRINCIPAL
1000 GET #2,E
1010 REM MOVER CURSOR
1020 IF E=87 THEN Y=Y-C:GOTO 1500
1030 IF E=88 THEN Y=Y+C:GOTO 1500
1040 IF E=68 THEN X=X+C:GOTO 1500
1050 IF E=65 THEN X=X-C:GOTO 1500
1060 IF E=69 THEN X=X+C:Y=Y-C:GOTO 150
0
1070 IF E=67 THEN X=X+C:Y=Y+C:GOTO 150
0
1080 IF E=81 THEN X=X-C:Y=Y-C:GOTO 150
0
1090 IF E=90 THEN X=X-C:Y=Y+C:GOTO 150
0
1100 REM ORDEN ENCONTRAR
1110 IF E=76 THEN GOTO 3000
1120 IF E=26 THEN GOTO 5000
1130 IF E=16 THEN GOTO 4000
1140 IF E=19 THEN GOTO 6000
1150 IF E=12 THEN GOTO 7000
1160 IF E=14 THEN CLOSE #2:CLR :RUN
1170 IF E=80 THEN GOTO 3100
1180 IF E=85 THEN GOTO 3110
1190 IF E=3 THEN PRINT "NUEVA LONGITUD
 DE PASO DEL CURSOR ";:INPUT C:PRINT :
PRINT :PRINT :GOTO 1000
1200 IF E=17 THEN GOTO 20000
1499 REM CONTROL DE ZONA
1500 IF X>XMAX THEN X=XMAX
1510 IF X<0 THEN X=0
1520 IF Y<0 THEN Y=0
```



```
1530 IF Y>YMAX THEN Y=YMAX
1540 GOSUB 2000
1545 PRINT "X:";X;" Y:";Y;" F:";F:IF E
=155 THEN GOTO 3150
1550 IF F1=0 THEN COLOR HF:PLOT X1,Y1
1559 REM COLOCAR PUNTO
1560 IF E=80 THEN F1=1:COLOR SF:PLOT X
,Y:PX(PZ)=X:PY(PZ)=Y:PC(PZ)=3:PZ=PZ+1
1570 IF E=32 THEN F1=0:COLOR HF:PLOT X
,Y:PX(PZ)=X:PY(PZ)=Y:PC(PZ)=2:PZ=PZ+1
1580 GOTO 1000
1999 REM ********** MOSTRAR POSICION
2000 LOCATE X,Y,F
2010 FOR I=1 TO 20:COLOR SF:PLOT X,Y:C
OLOR HF:PLOT X,Y:NEXT I
2020 COLOR HF:PLOT X,Y
2030 RETURN
2999 REM ********** ENTRADA LINEA
3000 IF DESPUES=0 THEN PRINT "LINEA DE
 X=";X;", Y=";Y:VX=X:VY=Y:DESPUES=-1:G
OTO 1000
3005 PRINT "DESPUES ";X;", ";Y:NX=X:NY
=Y:DESPUES=0
3010 COLOR SF:PLOT VX,VY:DRAWTO NX,NY:
IF REVISION=0 THEN E=74:GOTO 3030
3020 PRINT "O.K.?":GET #2,E
3030 IF E=74 THEN VX(LZ)=VX:VY(LZ)=VY:
NX(LZ)=NX:NY(LZ)=NY:PRINT :PRINT :LZ=L
Z+1:GOTO 1000
3040 IF E<>74 THEN COLOR HF:PLOT VX,VY
:DRAWTO NX,NY:GOTO 1000
3099 REM ********** ENTRADA PUNTO
3100 PRINT "X,Y ";:INPUT X,Y
3110 IF X>XMAX THEN PRINT "** X ES MUY
 ELEVADO":GOTO 3100
3120 IF Y>YMAX THEN PRINT "** Y ES MUY
 ELEVADO":GOTO 3100
3130 IF Y<0 THEN PRINT "** Y ES INADMI
SIBLE":GOTO 3100
3140 IF X<0 THEN PRINT "** X ES INADMI
```

```
SIBLE":GOTO 3100
3150 COLOR SF:PLOT X,Y:IF REVISION=0 T
HEN E=74:GOTO 3170
3160 PRINT "O.K.?":GET #2,E
3170 IF E=74 THEN PX(PZ)=X:PY(PZ)=Y:PR
INT :PRINT :PZ=PZ+1:GOTO 1000
3180 IF E<>74 THEN COLOR HF:PLOT X,Y:G
OTO 1000
3999 REM ************ CREAR PROGRAMAS
4000 PRINT "NOMBRE DE LA IMAGEN ";:INP
UT NOMBRE$:IF LEN(NOMBRE$)>8 THEN PRIN
T " ** MUY LARGO! **":GOTO 6000
4005 PRINT "PRIMER NUMERO DE LINEA ";:
INPUT LINEA
4010 NOMBRE$(LEN(NOMBRE$)+1)=".BAS"
4020 FILE$="D:":FILE$(LEN(FILE$)+1)=NO
MBRE$
4030 OPEN #1,8,0,FILE$
4050 Z$="GRAPHICS ":X=MODUS:GOSUB 80:H
$=":SETCOLOR 2,":GOSUB 40:X=HF:GOSUB 8
0:H$=",":GOSUB 40:X=HE:GOSUB 80:GOSUB
20
4060 Z$="COLOR ":X=SF:GOSUB 80:GOSUB 2
0
4100 FOR P=0 TO PZ-1
4110 Z$="PLOT ":X=PX(P):GOSUB 80:H$=",
":GOSUB 40:X=PY(P):GOSUB 80:GOSUB 20
4120 NEXT P
4150 FOR L=0 TO LZ-1
4160 Z$="PLOT ":X=VX(L):GOSUB 80:H$=",
":GOSUB 40:X=VY(L):GOSUB 80:H$="":DRAWT
O":GOSUB 40
4170 X=NX(L):GOSUB 80:H$=",":GOSUB 40:
X=NY(L):GOSUB 80:GOSUB 20
4180 NEXT L
4190 CLOSE #1:PRINT "IMAGEN ESTA PROGR
AMADA!":PRINT :GOTO 1000
4999 REM *** DIBUJAR IMAGEN NUEVAMENTE
5000 GRAPHICS MODUS:SETCOLOR 2,HF,HE:C
OLOR SF
```



```
5010 PZ=PZ-1:FOR P=0 TO PZ-1
5020 X=PX(P):Y=PY(P):PLOT X,Y
5030 NEXT P
5040 LZ=LZ-1:FOR L=0 TO LZ-1
5050 PLOT VX(L),VY(L):NX=NX(L):NY=NY(L
):DRAWTO NX,NY
5060 NEXT L
5070 PRINT "DIBUJO TERMINADO!":GOTO 10
00
5999 REM *** ALMACENAR DATOS DE IMAGEN
6000 PRINT "ALMACENAR: NOMBRE DE IMAGE
N";:INPUT NOMBRE$:IF LEN(NOMBRE$)>8 TH
EN PRINT " ** MUY LARGO! **":GOTO 6000
6010 NOMBRE$(LEN(NOMBRE$)+1)=".BDT"
6020 FILE$="D:":FILE$(LEN(FILE$)+1)=NO
MBRE$
6030 OPEN #1,8,8,FILE$
6040 PRINT #1,MODUS:PRINT #1,HF:PRINT
#1,HE:PRINT #1,PZ:PRINT #1,LZ
6050 FOR P=0 TO PZ-1
6060 PRINT #1,PX(P):PRINT #1,PY(P)
6070 NEXT P
6080 FOR L=0 TO LZ-1
6090 PRINT #1,VX(L):PRINT #1,VY(L):PRI
NT #1,NX(L),NY(L)
6100 NEXT L
6110 PRINT "DATOS DE IMAGEN ALMACENADO
S!":CLOSE #1:GOTO 1000
6999 REM ****** CARGAR DATOS DE IMAGEN
7000 PRINT "CARGAR:NOMBRE DE LA IMAGEN
 ";:INPUT NOMBRE$:IF LEN(NOMBRE$)>8 TH
EN PRINT " ** MUY LARGO! **":GOTO 6000
7010 NOMBRE$(LEN(NOMBRE$)+1)=".BDT"
7020 FILE$="D:":FILE$(LEN(FILE$)+1)=NO
MBRE$
7030 OPEN #1,4,0,FILE$
7040 INPUT #1,MODUS:INPUT #1,HF:INPUT
#1,HE:INPUT #1,PZ:INPUT #1,LZ
7050 FOR P=0 TO PZ-1
7060 INPUT #1,X:INPUT #1,Y:PX(P)=X:PY(
```

```
P)=Y
7070 NEXT P
7080 FOR L=0 TO LZ-1
7090 INPUT #1,X:INPUT #1,Y:VX(L)=X:VY(
L)=Y:INPUT #1,X:INPUT #1,Y:NX(L)=X:NY(
L)=Y
7100 NEXT L
7110 PRINT "DATOS DE IMAGEN CARGADOS!"
:CLOSE #1:GOTO 1000
7999 REM ********************* REVISION?
8000 IF REVISION=-1 THEN REVISION=0:PR
INT "ATENCION! NO CONSULTAR!":GOTO 100
0
8010 IF REVISION=0 THEN REVISION=-1:PR
INT "SE PRECISA CONFIRMACION.":GOTO 10
00
10000 PRINT "**** ERROR ****":GOTO 100
0
20000 CLOSE #1:CLOSE #2:GRAPHICS 0:END
```


GENERADOR DE AVENTURAS

GENERADOR DE AVENTURAS
'VENTUREFIX'

Este programa le confrontará con todas las preguntas que deben contestarse para crear un completo programa de aventuras capaz de funcionar.

Antes de ejecutar el 'Venturefix', usted ya debe haber definido la acción, confeccionado una lista de todos los lugares, objetos, palabras y mensajes, y formulado el texto de todas las acciones.

El programa no le evitará el trabajo creativo, sino sólo la rutina de la programación.

Ni siquiera necesitará conocer el significado de la instrucción PRINT y, no obstante, podrá crear programas Basic correctos.

Venturefix le pedirá en forma de diálogo uno tras otro todos los datos necesarios, y le informará continuamente sobre el estado de desarrollo del programa, hasta que se haya elaborado una aventura completa, según el estilo de este libro.

REFERENCIAS PARA POSEEDORES DE ATARI 600 XL

No se asuste al apreciar que el siguiente listado ocupa 30 - 40 kbytes, ya que Venturefix se puede descomponer en varias partes, permitiendole también a usted trabajar con este generador de programas.

Divida el siguiente listado en bloques que quepan justamente en la memoria de su ordenador (durante la entrada del programa, entre de vez en cuando PRINT FRE(0); en ningún

caso será preciso, que queden más de 1000 bytes disponibles).

No hace falta cambiar el número de línea; sólo debe procurar no separar el programa en medio de un bucle y no prescindir de los subprogramas necesarios en las líneas 20 a 100.

Además, la primera línea de todas los apartados después de la primera debe decir:

```
0 OPEN #1,9,0,"D:ADVENTUR.TXT"
```

Asimismo, finalice cada parte con:

```
60000 CLOSE #1:PRINT "SE CARGA LA PARTE
SIGUIENTE DEL PROGRAMA":RUN"D:VENTUREx
```

indicando, naturalmente, el número de la siguiente parte del programa en x.


```
1 REM 3.11.84
5 GRAPHICS 0:SETCOLOR 2,0,0:POSITION 1
,5
6 PRINT "VENTUREFIX":PRINT "(c) 1984 b
y Joerg Walkowiak":PRINT :PRINT :PRINT
 "del libro DATA BECKER:"
7 PRINT :PRINT "AVENTURAS, y como se p
rograman en el  ATARI":PRINT :PRINT
8 PRINT "(C) 1984 BY DATA BECKER, DUES
SELDORF":FOR I=1 TO 3000:NEXT I
9 DIM AVENTURA$(12),ENTRADA$(20),COPYR
IGHT$(35),FECHA$(10),Z$(120),H$(80),VE
RSION$(5),VERBOS$(80),MO$(22),M1$(38)
10 DIM M$(40):GOTO 100
20 IF IMPRESION THEN LPRINT NUMEROLINE
A;" ";Z$
30 PRINT #1,NUMEROLINEA,Z$:NUMEROLINEA
=NUMEROLINEA+SALTOLINEAS:RETURN
40 Z$(LEN(Z$)+1)=H$:RETURN
50 GOSUB 70:GOSUB 40:GOSUB 70:H$=":RET
URN":GOSUB 40:GOSUB 20:RETURN
60 GOSUB 40:GOSUB 70:GOSUB 20:RETURN
70 Z$(LEN(Z$)+1)=CHR$(34):RETURN
80 Z$(LEN(Z$)+1)=STR$(X):RETURN
90 OPEN #2,4,0,"K:":GET #2,ENTRADA:CLO
SE #2:RETURN
95 GOSUB 40:GOSUB 20:RETURN
100 FOR I=1 TO 300:NEXT I:GRAPHICS 0:S
ETCOLOR 2,0,0
110 AVENTURA$="D:":PRINT "COMO SE LLAM
ARA LA AVENTURA:":PRINT :INPUT ENTRADA
$
120 IF LEN(ENTRADA$)>8 THEN PRINT "POR
 FAVOR, UN MAXIMO DE 8 LETRAS!":GOTO 1
00
130 AVENTURA$(LEN(AVENTURA$)+1)=ENTRAD
A$:AVENTURA$(LEN(AVENTURA$)+1)=".ADV"
140 PRINT :PRINT "DEBE CREARSE A LA VE
Z UN LISTADO EN":PRINT "UNA IMPRESORA?
"
```

```
150 GOSUB 90
160 IF (ENTRADA=74 OR ENTRADA=202) THE
N IMPRESION=-1
170 NUMEROLINEA=10
180 SALTOLINEA=5
200 GRAPHICS 0:SETCOLOR 2,0,0:PRINT "D
ATOS PARTICULARES DE LA AVENTURA: ";AV
ENTURA$(1,7)
210 PRINT "-------------------------
----------"
220 PRINT "NOMBRE DEL AUTOR ":INPUT CO
PYRIGHT$:IF LEN(COPYRIGHT$)>35 THEN PR
INT "... DEMASIADO LARGO!":GOTO 220
230 PRINT ENTRADA$;", VERSION$: ":INPU
T VERSION$
240 PRINT "FECHA DE HOY ":INPUT FECHA$
250 PRINT "CUANTOS LUGARES ":INPUT AR:
DIM PA(AR,6)
260 PRINT "CUANTOS OBJETOS ":INPUT AO:
DIM OB(AO)
270 PRINT "CUANTOS VERBOS ":INPUT AV
280 PRINT "VALOR ESTIMADO, CUANTAS BAN
DERAS ":INPUT AF
290 PRINT :PRINT "EN QUE LUGAR DEBE CO
MENZAR ";AVENTURA$(3,10):INPUT JUGADOR
500 GRAPHICS 0:PRINT "PARTE  I :INICIA
LIZACION":PRINT :PRINT
520 PRINT " - ABRIR FICHERO":OPEN #1,8
,0,AVENTURA$
530 PRINT " - ENCABEZAMIENTO DE PROGRA
MA":Z$="REM ":Z$(LEN(Z$)+1)=ENTRADA$:G
OSUB 20
540 Z$="REM VERSION ":Z$(LEN(Z$)+1)=VE
RSION$:GOSUB 20
550 Z$="REM (C) BY ":Z$(LEN(Z$)+1)=COP
YRIGHT$:GOSUB20
560 Z$="REM CONFECCIONADO EL ":Z$(LEN(
Z$)+1)=FECHA$:GOSUB 20
570 Z$="REM CON AYUDA DE:":GOSUB 20
580 Z$="REM  VENTUREFIX              ":G
OSUB 20
590 Z$="REM  (C) 1984  BY WALKOWIAK":G
```



```
OSUB 20
600 PRINT " - TITULO":Z$="REM---------
----------------------":GOSUB 20
610 Z$="GRAPHICS 0:SETCOLOR 2,1,0:SETC
OLOR 4,1,0":GOSUB 20
620 Z$="PRINT ":GOSUB 70:Z$(LEN(Z$)+1)
=ENTRADA$:GOSUB 20
630 Z$="REM ***** INSERTAR IMAGEN TITU
LAR":GOSUB 20:NUMEROLINEA=NUMEROLINEA+
50
640 PRINT " - DATOS PARTICULARES":Z$="
AR=":X=AR:GOSUB 80:GOSUB 20
650 Z$="AO=":X=AO:GOSUB 80:GOSUB 20
660 Z$="AF=":X=AF:GOSUB 80:GOSUB 20
670 Z$="SP=":X=JUGADOR:GOSUB 80:GOSUB
20
680 Z$="TRAP 4900":GOSUB 20
700 PRINT " - DIMENSIONAR CAMPOS"
710 Z$="DIM LU$(80),PA(AR,6),DI$(36),E
NTRADA$(20),T$(40),OB(AO),LZ$(38),DZ$(
50),ZE$(3),FL(AF),OK$(4)":GOSUB 20
720 Z$="DIM VERBOS$(120),EV$(20),EO$(2
0),OBJETOS$(300),MO$(40)":GOSUB 20
730 Z$="OK$=":GOSUB 70:H$="O.K.":GOSUB
 60
800 GRAPHICS 0:SETCOLOR 2,0,0:PRINT "P
ARTE  II : DATOS DE LA AVENTURA":PRINT
 :PRINT
805 Z$="VERBOS$=":GOSUB 70
810 PRINT "CUANTAS LETRAS DEBEN":PRINT
 "SER SIGNIFICATIVAS ";:INPUT WL
820 PRINT :PRINT "NECESITA MAS QUE OTR
OS 80 CARACTERES?":GOSUB 90:OTRA=0
830 IF (ENTRADA=74 OR ENTRADA=202) THE
N OTRA=-1
840 PRINT :PRINT "AHORA ENTRE SU VERBO
S$:"
850 INPUT H$
860 IF LEN(H$)>80 THEN GOTO 800
870 GOSUB 40:GOSUB 70:GOSUB 20
```

```
880 IF  NOT OTRA THEN GOTO 900
890 Z$="VERBOS$(LEN(VERBOS$)+1)=":GOSU
B 70:GOTO 820
900 Z$="OBJETOS$=":GOSUB 70
920 PRINT :PRINT "NECESITA MAS DE 80 C
ARACTERES":PRINT "PARA LOS OBJETOS?":G
OSUB 90:OTRA=0
930 IF (ENTRADA=74 OR ENTRADA=202) THE
N OTRA=-1
940 PRINT :PRINT "AHORA ENTRE SU OBJET
OS$:"
950 INPUT H$
960 IF LEN(H$)>80 THEN GOTO 900
970 GOSUB 40:GOSUB 70:GOSUB 20
980 IF  NOT OTRA THEN GOTO 1000
990 Z$="VERBOS$(LEN(VERBOS$)+1)=":GOSU
B 70:GOTO 920
1000 Z$="GOTO 700":GOSUB 20
1010 Z$="REM ****** DESCRIPCIONES DE L
UGAR":SALTOLINEA=1:NUMEROLINEA=200:GOS
UB 20
1020 GRAPHICS 0:SETCOLOR 2,0,0:PRINT "
PARTE II: ENTRAR LUGARES":PRINT :PRINT

1030 FOR I=1 TO AR
1040 H$="":PRINT "ESTOY ";:INPUT H$
1045 PRINT "Este lugar conduce":PRINT
" en el NORTE a lugar ";:INPUT D:PA(I,
1)=D
1050 PRINT " en el SUR a lugar   ";:IN
PUT D:PA(I,2)=D
1055 PRINT " en el OESTE a lugar ";:IN
PUT D:PA(I,3)=D
1060 PRINT " en el ESTE a lugar  ";:IN
PUT D:PA(I,4)=D
1065 PRINT " ARRIBA a lugar      ";:IN
PUT D:PA(I,5)=D
1070 PRINT " ABAJO a lugar       ";:IN
PUT D:PA(I,6)=D
1080 Z$="PRINT ":GOSUB 50
```



```
1090 NEXT I
1100 NUMEROLINEA=NUMEROLINEA+50
1110 Z$="REM ********************* OBJE
TOS":NUMEROLINEA=300:GOSUB 20
1120 GRAPHICS 0:SETCOLOR 2,0,0:PRINT "
PARTE II : ENTRAR OBJETOS":PRINT ;PRIN
T
1130 FOR I=1 TO AO
1140 PRINT "VEO ";:INPUT H$
1150 PRINT :PRINT "LUGAR INICIAL ";:IN
PUT SR:OB(I)=SR
1160 Z$="T$=":GOSUB 50
1170 NEXT I
1190 NUMEROLINEA=NUMEROLINEA+50
1200 GRAPHICS 0:PRINT "PARTE  II : DAT
OS DE LA AVENTURA":PRINT :PRINT
1210 PRINT " - EMPLAZAR OBJETOS"
1220 I=1
1230 Z$="DATA "
1240 H$=STR$(OB(I))
1250 I=I+1
1260 GOSUB 40
1270 IF LEN(H$)>100 THEN 1290
1280 IF I<AO+1 THEN H$=",":GOSUB 40:GO
TO 1240
1290 GOSUB 20:IF I<AO+1 THEN 1230
1300 PRINT " - CONECTAR LUGARES"
1310 Z$="DATA ":KOMMA=0
1320 FOR LU=1 TO AR
1330 FOR DI=1 TO 6
1340 IF KOMMA THEN H$=",":GOSUB 40
1350 H$=STR$(PA(LU,DI)):GOSUB 40:KOMMA
=-1
1360 IF LEN(Z$)>100 THEN GOSUB 20:Z$="
DATA ":KOMMA=0
1370 NEXT DI
1380 NEXT LU
1390 IF LEN(Z$)>1 THEN GOSUB 20
1400 PRINT " - TITULO : INSTRUCCIONES"
:Z$="REM *** AQUI INTRODUCIR TITULO 2"
```

```
:NUMEROLINEA=699:GOSUB 20
1410 Z$="PRINT :PRINT :PRINT ":GOSUB 7
0:H$="DESEA RECIBIR CONSEJOS PARA CONT
INUAR"
1420 GOSUB 40:GOSUB 70:H$=";:INPUT ENT
RADA$":GOSUB 40:NUMEROLINEA=700:GOSUB
20
1430 Z$="IF ENTRADA$(1,1)=":GOSUB 70:H
$="S":GOSUB 40:GOSUB 70:H$="THEN GOSUB
800":GOSUB 40:NUMEROLINEA=798:GOSUB 20
1440 Z$="GOTO 900":NUMEROLINEA=799:GOS
UB 20:SALTOLINEA=5
1450 Z$="GRAPHICS 0:SETCOLOR 2,1,0:SET
COLOR 4,1,0":GOSUB 20
1460 Z$="PRINT ":GOSUB 70:H$="
   ATARI - VENTURES          ":GOSUB 60
1470 Z$="PRINT ":GOSUB 70:H$="      (C)
 1984 BY JOERG WALKOWIAK     ":GOSUB 40
:GOSUB 20
1480 Z$="PRINT :PRINT ":GOSUB 70:H$="I
maginese un robot al que puede con-  t
rolar con numerosos comandos."
1490 GOSUB 60
1500 Z$="PRINT ":GOSUB 70:H$="Yo soy e
se robot y me pondre en su    lugar pa
ra enfrentarme a todos los":GOSUB 60
1510 Z$="PRINT ":GOSUB 70:H$="peligros
 de las aventuras mas exci-   tantes."
:GOSUB 60
1520 Z$="PRINT ":GOSUB 70:H$="Para que
 pueda manejarme comodamente, le descr
ibire en cada momento la si-":GOSUB 60
1530 Z$="PRINT ":GOSUB 70:H$="tuacion
en que me encuentre.":GOSUB 60
1540 Z$="PRINT ":GOSUB 70:H$="A contin
uacion, Ud me indicara con":GOSUB 60
1550 Z$="PRINT ":GOSUB 70:H$="dos pala
bras, como por ejemplo":GOSUB 60
1560 Z$="PRINT ":GOSUB 70:H$="INSPECCI
ONA PUERTA, COGE CUCHILLO,":GOSUB 60
```



```
1570 Z$="PRINT ":GOSUB 70:H$=" lo que
deba hacer.":GOSUB 60
1580 Z$="PRINT :PRINT ":GOSUB 70:H$="A
demas entiendo las ordenes:":GOSUB 60
1590 Z$="PRINT ":GOSUB 70:H$="
   SAVE, LOAD,":GOSUB 60
1600 Z$="PRINT ":GOSUB 70:H$="
 INVENTARIO y FIN.":GOSUB 60
1610 Z$="PRINT :PRINT ":GOSUB 70:H$="P
OR FAVOR, PULSE <RETURN> Y ...":GOSUB
40:GOSUB 70
1620 H$=";:INPUT ENTRADA$:RETURN":GOSU
B 40:GOSUB 20
1700 PRINT " - PARTE DE INICIALIZACION
":NUMEROLINEA=900:SALTOLINEA=10
1710 Z$="FOR I=1 TO AO:READ SITUACION
OB(I)=SITUACION:NEXT I":GOSUB 20
1720 Z$="FOR I=1 TO AF:FL(I)=0:NEXT I"
:GOSUB 20
1730 Z$="FOR LU=1 TO AR:FOR DI=1 TO 6"
:GOSUB 20
1740 Z$="READ META:PA(LU,DI)=META":GOS
UB 20
1750 Z$="NEXT DI:NEXT LU":GOSUB 20
1800 GRAPHICS 0:PRINT "PARTE III: ESCR
IBIR MOTOR DE AVENTURA":PRINT :PRINT
1810 NUMEROLINEA=1000
1820 Z$="GRAPHICS 0:SETCOLOR 2,1,0:SET
COLOR 4,1,0:GOTO 1030":GOSUB 20
1830 Z$="REM ******* TEXTO CLARO: DIRE
CCIONES":SALTOLINEA=1:NUMEROLINEA=1020
:PRINT " - DIRECCIONES":GOSUB 20
1840 Z$="T$=":GOSUB 70:H$="NORTE, ":GO
SUB 40:GOSUB 70:H$=":RETURN":GOSUB 40:
GOSUB 20
1850 Z$="T$=":GOSUB 70:H$="SUR, ":GOSU
B 40:GOSUB 70:H$=":RETURN":GOSUB 40:GO
SUB 20
1860 Z$="T$=":GOSUB 70:H$="OESTE, ":GO
SUB 40:GOSUB 70:H$=":RETURN":GOSUB 40:
```

```
GOSUB 20
1870 Z$="T$=":GOSUB 70:H$="ESTE, ":GOS
UB 40:GOSUB 70:H$=":RETURN":GOSUB 40:G
OSUB 20
1880 Z$="T$=":GOSUB 70:H$="ARRIBA, ":G
OSUB 40:GOSUB 70:H$=":RETURN":GOSUB 40
:GOSUB 20
1890 Z$="T$=":GOSUB 70:H$="ABAJO, ":GO
SUB 40:GOSUB 70:H$=":RETURN":GOSUB 40:
GOSUB 20
1900 NUMEROLINEA=1030:SALTOLINEA=10:Z$
="LZ$=":GOSUB 70:H$="
                    ":GOSUB 60
1910 Z$="T$=":GOSUB 70:GOSUB 70:GOSUB
20
1920 Z$="ZE$=CHR$(30):ZE$(LEN(ZE$)+1)=
CHR$(30):ZE$(LEN(ZE$)+1)=":GOSUB 70:H$
=".":GOSUB 60
1930 NUMEROLINEA=1080:PRINT " - SALIDA
 EN PANTALLA"
1940 Z$="PRINT":GOSUB 20
1950 Z$="FOR Z=0 TO 10:POSITION 2,Z:PR
INT LZ$:NEXT Z":GOSUB 20
1960 Z$="POSITION 2,0:PRINT ":GOSUB 70
:H$="ESTOY ":GOSUB 40:GOSUB 70:H$=";:G
OSUB 200+JU":GOSUB 40:GOSUB 20
1970 Z$="DZ$=":GOSUB 70:H$="VEO ":GOSU
B 60
1980 Z$="FOR I=1 TO A0":GOSUB 20
1990 Z$="IF OB(I)<>JU THEN GOTO 1170":
GOSUB 20
2000 Z$="GD=-1:GOSUB 300+I:T$(LEN(T$)+
1)=":GOSUB 70:H$=", ":GOSUB 60
2010 Z$="IF LEN(DZ$)+LEN(T$)>=38 THEN
PRINT DZ$:DZ$=T$(1,LEN(T$)):T$=":GOSUB
 70:GOSUB 70:H$=":GOTO 1170"
2020 GOSUB 40:GOSUB 20
2030 Z$="DZ$(LEN(DZ$)+1)=T$:T$=":GOSUB
 70:GOSUB 70:GOSUB 20
2040 Z$="NEXT I":GOSUB 20
```



```
2050 Z$="IF NOT GD THEN DZ$(LEN(DZ$)+1
)=":GOSUB 70:H$="NADA ESPECIAL. ":GOSU
B 60
2060 Z$="DZ$(LEN(DZ$)+1)=ZE$":GOSUB 20
2070 Z$="PRINT DZ$":GOSUB 20
2080 Z$="PRINT LZ$":GOSUB 20
2090 Z$="DZ$=":GOSUB 70:H$="PUEDO IR H
ACIA ":GOSUB 40:GOSUB 70:H$=":GD=0":GO
SUB 40:GOSUB 20
2100 Z$="FOR DI=1 TO 6":GOSUB 20
2110 Z$="IF PA(JU,DI)=0 THEN GOTO 1280
":GOSUB 20
2120 Z$="GOSUB 1020+DI:GD=-1":GOSUB 20
2130 Z$="IFLEN(DZ$)+LEN(T$)>=38THENPRI
NTDZ$:DZ$=T$(1,LEN(T$)):T$=":GOSUB 70:
GOSUB 70:H$=":GOTO1280":GOSUB 40:GOSUB
 20
2140 Z$="DZ$(LEN(DZ$)+1)=T$:T$=":GOSUB
 70:GOSUB 70:GOSUB 20
2142 Z$="NEXT DI":GOSUB 20
2144 Z$="IF NOT GD THEN DZ$(LEN(DZ$)+1
)=":GOSUB 70:H$="NINGUN SITIO":GOSUB 6
0
2146 Z$="DZ$(LEN(DZ$)+1)=ZENDE$"
2150 Z$="PRINT DZ$":GOSUB 20
2160 Z$="PRINT ":GOSUB 70:H$="--------
--------------------------------":GOSUB
 60
2170 Z$="REM *** AQUI PRUEBA FIN DE JU
EGO ***":GOSUB 20
2180 Z$="REM *** O TRAMPAS ASI COMO **
*******":GOSUB 20
2190 Z$="REM *** TRAMPAS TAMBIEN EN BA
SE 1399":GOSUB 20
2200 PRINT " - MOVER JUGADOR":NUMEROLI
NEA=1390
2210 Z$="POSITION2,23:PRINT ":GOSUB 70
:H$="QUE DEBO HACER ":GOSUB 40:GOSUB 7
0:H$=";:INPUTENTRADA$":GOSUB 40:GOSUB
20
```

```
2220 Z$="IF LEN(ENTRADA$)>2 THEN 1480"
2230 Z$="IF ENTRADA$=":GOSUB 70:H$="N"
:GOSUB 40:GOSUB 70:H$="AND PA(JU,1)<>0
 THEN JU=PA(JU,1):PRINT OK$:GOTO 1080"
2240 GOSUB 95
2250 Z$="IF ENTRADA$=":GOSUB 70:H$="S"
:GOSUB 40:GOSUB 70:H$="AND PA(JU,2)<>0
 THEN JU=PA(JU,2):PRINT OK$:GOTO 1080"
2260 GOSUB 95
2270 Z$="IF ENTRADA$=":GOSUB 70:H$="O"
:GOSUB 40:GOSUB 70:H$="AND PA(JU,3)<>0
 THEN JU=PA(JU,3):PRINT OK$:GOTO 1080"
2280 GOSUB 95
2290 Z$="IF ENTRADA$=":GOSUB 70:H$="E"
:GOSUB 40:GOSUB 70:H$="AND PA(JU,4)<>0
 THEN JU=PA(JU,4):PRINT OK$:GOTO 1080"
2300 GOSUB 95
2310 Z$="IF ENTRADA$=":GOSUB 70:H$="AR
":GOSUB 40:GOSUB 70:H$="AND PA(JU,5)<>
OTHEN JU=PA(JU,5):PRINT OK$:GOTO 1080"
2320 GOSUB 95
2330 Z$="IF ENTRADA$=":GOSUB 70:H$="AB
":GOSUB 40:GOSUB 70:H$="AND PA(JU,6)<>
OTHEN JU=PA(JU,6):PRINT OK$:GOTO 1080"
2340 GOSUB 95
2345 Z$="IFLEN(ENTRADA$)<3THENPRINT":G
OSUB 70:H$="HACIA ALLI NINGUN CAMINO!"
:GOSUB 40:GOSUB 70:H$=":GOTO1080":GOSU
B 95
2350 Z$="IF LEN(ENTRADA$)>8 THEN GOTO
2000":GOSUB 20
2400 PRINT " - INVENTARIO":NUMEROLINEA
=1500:SALTOLINEA=5
2410 Z$="IF ENTRADA$(1,3)<>":GOSUB 70:
H$="INV":GOSUB 40:GOSUB 70:H$=" THEN G
OTO 1600":GOSUB 95
2420 Z$="PRINT ":GOSUB 70:H$="LLEVO CO
NMIGO LO SIGUIENTE:":GOSUB 60
2430 Z$="FOR I=1 TO AO":GOSUB 20
2440 Z$="IF OB(I)=-1 THEN GOSUB 300+I:
```



```
PRINT T$":GOSUB 20
2450 Z$="NEXT I":GOSUB 20
2460 Z$="GOTO 1080":GOSUB 20
2500 PRINT " - SAVE GAME":NUMEROLINEA=
1600
2510 Z$="IF ENTRADA$(1,3)<>":GOSUB 70:
H$="SAV":GOSUB 40:GOSUB 70:H$=" THEN G
OTO 1700":GOSUB 40:GOSUB 20
2520 Z$="PRINT ":GOSUB 70:H$="CON QUE
NOMBRE ":GOSUB 40:GOSUB 70:H$=";:INPUT
ENTRADA$:IFLEN(ENTRADA$)>8 THEN PRINT"
2530 GOSUB 40:GOSUB 70:H$="UN POCO MAS
 CORTO!":GOSUB 40:GOSUB 70:H$=":GOTO 1
605":GOSUB 95
2540 Z$="T$=":GOSUB 70:H$="D:":GOSUB 4
0:GOSUB 70:H$=":ENTRADA$(LEN(ENTRADA$)
+1)=":GOSUB 40:GOSUB 70:H$=".DAT":GOSU
B 60
2550 Z$="T$(LEN(T$)+1)=ENTRADA$":GOSUB
 20
2560 Z$="OPEN #1,8,0,T$":GOSUB 20
2570 Z$="PRINT #1,JU":GOSUB 20
2580 Z$="FOR I=1 TO AO:PRINT #1,OB(I):
NEXT I":GOSUB 20
2590 Z$="FOR LU=1 TO AR":GOSUB 20
2600 Z$="FOR DI=1 TO 6":GOSUB 20
2610 Z$="PRINT #1,PA(LU,DI)"
2620 Z$="NEXT DI":GOSUB 20
2630 Z$="NEXT LU":GOSUB 20
2640 Z$="FOR I=1 TO AF:PRINT #1,FL(I):
NEXT I":GOSUB 20
2650 Z$="CLOSE #1:PRINT OK$:GOTO 1080"
:GOSUB 20
2700 PRINT " - LOAD GAME":NUMEROLINEA=
1700
2710 Z$="IF ENTRADA$(1,3)<>":GOSUB 70:
H$="LOA":GOSUB 40:GOSUB 70:H$=" THEN G
OTO 1900":GOSUB 95
2720 Z$="PRINT ":GOSUB 70:H$="QUE JUEG
O ":GOSUB 40:GOSUB 70:H$=";:INPUT ENTR
```

```
ADA$:IF LEN(ENTRADA$)>8 THEN PRINT"
2730 GOSUB 40:GOSUB 70:H$="ESTE NO PUE
DE EXISTIR!":GOSUB 40:GOSUB 70:H$=":GO
TO 1705":GOSUB 95
2740 Z$="T$=":GOSUB 70:H$="D:":GOSUB 4
0:GOSUB 70:H$=":ENTRADA$(LEN(ENTRADA$)
+1)=":GOSUB 40:GOSUB 70:H$=".DAT":GOSU
B 60
2750 Z$="T$(LEN(T$)+1)=ENTRADA$":GOSUB
 20
2760 Z$="OPEN #1,4,0,T$":GOSUB 20
2770 Z$="INPUT #1,JU":GOSUB 20
2780 Z$="FOR I=1 TO AO:INPUT #1,LO:OB(
I)=LO:NEXT I":GOSUB 20
2790 Z$="FOR LU=1 TO AR":GOSUB 20
2800 Z$="FOR DI=1 TO 6":GOSUB 20
2810 Z$="INPUT #1,META:PA(LU,DI)=META"
:GOSUB 20
2820 Z$="NEXT DI":GOSUB 20
2830 Z$="NEXT LU":GOSUB 20
2840 Z$="FOR I=1 TO AF:INPUT #1,META:F
L(I)=META:NEXT I":GOSUB 20
2850 Z$="CLOSE #1:PRINT OK$:GOTO 1080"
:GOSUB 20
2900 PRINT " - INS y FIN":NUMEROLINEA=
1900
2910 Z$="IF ENTRADA$(1,3)<>":GOSUB 70:
H$="INS":GOSUB 40:GOSUB 70:H$=" THEN G
OTO 1910":GOSUB 95
2920 Z$="GOSUB 800:GOTO 1080":GOSUB 20
2930 Z$="IF ENTRADA$(1,3)<>":GOSUB 70:
H$="FIN":GOSUB 40:GOSUB 70:H$=" THEN G
OTO 2000":GOSUB 95
2940 Z$="GRAPHICS 0:PRINT ":GOSUB 70:H
$="EL AUTOR LE DESEA MAS SUERTE EN EL"
:GOSUB 40:GOSUB 70:H$=":PRINT ":GOSUB
40
3000 GOSUB 70:H$="PROXIMO INTENTO!":GO
SUB 40:GOSUB 70:H$=":PRINT :PRINT :PRI
NT :END":GOSUB 95
```



```
3010 PRINT " - ANALISIS DE ENTRADA":NU
MEROLINEA=2000:SALTOLINEA=10
3020 Z$="LN=LEN(ENTRADA$)":GOSUB 20
3030 Z$="FOR I=1 TO LN":GOSUB 20
3040 Z$="IF ENTRADA$(I,I)<>":GOSUB 70:
H$=" ":GOSUB 40:GOSUB 70:H$=" THEN NEX
T I":GOSUB 95
3050 Z$="EV$=ENTRADA$(1,I)":GOSUB 20
3060 Z$="IF LEN(EV$)=LN-1 THEN GOTO 20
90":GOSUB 20
3070 Z$="EO$=ENTRADA$(I+1,LN)":GOSUB 2
0
3080 Z$="VN=0:N=0":GOSUB 20
3090 Z$="EV$=EV$(1,":X=WL:GOSUB 80:H$=
")":GOSUB 40:GOSUB 20
3100 Z$="EO$=EO$(1,":X=WL:GOSUB 80:H$=
")":GOSUB 40:GOSUB 20
3110 Z$="FOR I=1 TO LEN(VERBOS$) STEP
":GOSUB 80:GOSUB 20
3120 Z$="VN=VN+1":GOSUB 20
3130 X=WL-1:Z$="IF VERBOS$(I,I+":GOSUB
 80:H$=")=EV$ THEN 2140":GOSUB 95
3140 Z$="NEXT I":GOSUB 20
3150 Z$="PRINT ":GOSUB 70:H$="NO ENTIE
NDO EL VERBO!":GOSUB 40:GOSUB 70:H$=":
GOTO 1080":GOSUB 95
3160 X=WL:Z$="FOR I=1 TO LEN(OBJETOS$)
 STEP ":GOSUB 80:GOSUB 20
3170 Z$="N=N+1":GOSUB 20
3180 X=WL-1:Z$="IF OBJETOS$(I,I+":GOSU
B 80:H$=")=EO$ THEN 2200":GOSUB 95
3190 Z$="NEXT I":GOSUB 20
3200 Z$="PRINT ":GOSUB 70:H$="NO ENTIE
NDO EL OBJETO!":GOSUB 40:GOSUB 70:H$="
:GOTO 1080":GOSUB 95
3210 PRINT " - TABLA DE SALTOS":NUMERO
LINEA=2200:X=4000:Z$="ON VN GOTO"
3220 FOR I=1 TO AV
3230 X=X+1000:GOSUB 80
3235 IF I<>AV THEN H$=",":GOSUB 40
```

```
3240 NEXT I
3245 GOSUB 20
3250 PRINT " - TITULO: JUGADOR MUERTO"
:NUMEROLINEA=4500
3260 Z$="GRAPHICS 0":GOSUB 20
3270 Z$="PRINT ":GOSUB 70:H$="LO QUE F
ALTABA!":GOSUB 70:H$=":PRINT :PRINT MO
$":GOSUB 95
3280 Z$="PRINT :PRINT ":GOSUB 70:H$="E
STOY MUERTO!":GOSUB 40:GOSUB 70:H$=":P
RINT":GOSUB 95
3290 Z$="PRINT ":GOSUB 70:H$="QUIERE Q
UE LO INTENTE OTRA VEZ ":GOSUB 40:GOSU
B 70:H$=";:INPUT ENTRADA$":GOSUB 95
3300 Z$="IF ENTRADA$(1,1)=":GOSUB 70:H
$="S":GOSUB 40:GOSUB 70:H$=" THEN CLR
:GOTO 100":GOSUB 95
3310 Z$="GOTO 1960":GOSUB 20
3320 PRINT " - TITULO: VICTORIA":NUMER
OLINEA=4800
3330 Z$="GRAPHICS 0":GOSUB 20
3340 Z$="PRINT ":GOSUB 70:H$="ENHORABU
ENA!":GOSUB 60
3350 Z$="PRINT :PRINT :PRINT ":GOSUB 7
0:H$="HA RESUELTO EL PROBLEMA PLANTEAD
O Y":GOSUB 40:GOSUB 20
3360 Z$="PRINT :PRINT ":GOSUB 70:H$="P
UEDE AFRONTAR UNA NUEVA AVENTURA":GOSU
B 40:GOSUB 20
3380 Z$="PRINT :PRINT :PRINT :END":GOS
UB 20
3390 NUMEROLINEA=4900:Z$="PRINT ":GOSU
B 70:H$="ATENCION ERROR!":GOSUB 40:GOS
UB 70:H$=":GOTO 1080":GOSUB 95
4000 M$="PARTE IV: CONDICIONES & ACCIO
NES":SALTOLINEA=1:Z1=5000:NUMEROLINEA=
Z1:O=0:V=0
4005 V=V+1
4010 O=O+1
4015 X=0:Z$="IF N=":GOSUB 80:H$=" AND
```



```
":GOSUB 40
4020 MO$="PARTE IV: CONDICIONES":M1$="
"
4030 GRAPHICS 0:SETCOLOR 2,0,0:PRINT M
O$:PRINT M1$
4040 PRINT " 1 - OBJETO ESTA EN LUGAR"
4050 PRINT " 2 - OBJETO NO ESTA EN LUG
AR"
4060 PRINT " 3 - JUGADOR TIENE OBJETO"
4070 PRINT " 4 - FLAG ACTIVADO"
4080 PRINT " 5 - FLAG DESACTIVADO"
4090 PRINT " 6 - JUGADOR DEBE ESTAR EN
 LUGAR:"
4094 PRINT " 7 - Y OTRA CONDICION"
4097 PRINT " 8 - U OTRA CONDICION"
4100 PRINT :PRINT " 0 - CONDICION O.K.
"
4110 PRINT M1$
4120 PRINT "ELIJA VERBO ";V;" OBJETO "
;O
4130 INPUT E
4140 IF E=1 THEN H$="OB(N)=JU ":GOSUB
40
4150 IF E=2 THEN H$="OB(N)<>JU ":GOSUB
 40
4160 IF E=3 THEN PRINT "OBJETO NUMERO
":INPUT X:H$="OB(":GOSUB 40:GOSUB 80:H
$=")=JU":GOSUB 40
4170 IF E=4 THEN PRINT "FLAG NUMERO ":
INPUT X:H$="FL(":GOSUB 40:GOSUB 80:H$=
")=-1":GOSUB 40
4180 IF E=5 THEN PRINT "FLAG NUMERO ":
INPUT X:H$="FL(":GOSUB 40:GOSUB 80:H$=
")<>-1":GOSUB 40
4190 IF E=6 THEN PRINT "QUE LUGAR ":IN
PUT X:H$="JU=":GOSUB 40
4200 IF E=7 THEN H$=" AND ":GOSUB 40
4210 IF E=8 THEN H$=" OR ":GOSUB 40
4220 IF E<>0 THEN 4030
4225 H$="THEN":GOSUB 40
```

```
4300 MO$="PARTE IV: ACCIONES"
4310 GRAPHICS 0:SETCOLOR 2,0,0:PRINT M
O$:PRINT M1$
4320 PRINT " 1 - OBJETO DESAPARECE"
4330 PRINT " 2 - OBJETO AL INVENTARIO"
4340 PRINT " 3 - OBJETO APARECE DE NUE
VO"
4350 PRINT " 4 - FLAG ES ACTIVADO"
4360 PRINT " 5 - FLAG ES DESACTIVADO"
4370 PRINT " 6 - ABRIR PASADA"
4380 PRINT " 7 - EMITIR MENSAJE"
4390 PRINT " 8 - JUGADOR MUERE"
4400 PRINT " 9 - JUGADOR VENCE"
4410 PRINT :PRINT " 0 - ACCIONES O.K."
4420 PRINT M1$:PRINT "ELIJA VERBO ";V;
"OBJETO ";O:INPUT E
4430 IF E=1 THEN PRINT "OBJETO NUMERO
";:INPUT X:H$="OB(":GOSUB 40:GOSUB 80:
H$=")=0":GOSUB 40
4440 IF E=2 THEN PRINT "OBJETO NUMERO
";:INPUT X:H$="OB(":GOSUB 40:GOSUB 80:
H$=")=-1":GOSUB 40
4450 IF E=3 THEN PRINT "OBJETO NUMERO
";:INPUT X:H$="OB(":GOSUB 40:GOSUB 80:
H$=")=JU":GOSUB 40
4460 IF E=4 THEN PRINT "FLAG NUMERO ";
:INPUT X:H$="FL(":GOSUB 40:GOSUB 80:H$
=")=-1":GOSUB 40
4470 IF E=5 THEN PRINT "FLAG NUMERO ";
:INPUT X:H$="FL(":GOSUB 40:GOSUB 80:H$
=")=0":GOSUB 40
4480 IF E=6 THEN H$="PA(":GOSUB 40:PRI
NT "DE LUGAR ";:INPUT X:GOSUB 80:H$=",
":GOSUB 40:PRINT "EN DIRECCION ";:INPU
T X
4485 IF E=6 THEN GOSUB 80:H$=")=":GOSU
B 40:PRINT "A LUGAR ";:INPUT X:GOSUB 8
0
4490 IF E=7 THEN H$="PRINT":GOSUB 40:G
OSUB 70:PRINT "TEXT ";:INPUT H$:GOSUB
```



```
40:GOSUB 70
4500 IF E=8 THEN H$="MO$=":GOSUB 70:PR
INT "TEXT ";:INPUT H$:GOSUB 40:GOSUB 7
0:H$="GOTO 4500":GOSUB 40
4510 IF E=9 THEN H$="GOTO 4800":GOSUB
40
4520 IF E<>0 THEN H$=":":GOSUB 40:GOTO
 4310
4525 H$="GOTO 1080":GOSUB 95
4527 PRINT "MAS REFERENTE AL MISMO OBJ
ETO?":GOSUB 90:IF ENTRADA=74 THEN GOTO
 4015
4530 IF O<>AO THEN GOTO 4010
4540 O=0:Z1=Z1+1000:NUMEROLINEA=Z1
4550 IF V<>AV THEN GOTO 4005
10000 CLOSE #1:GRAPHICS 0:SETCOLOR 2,0
,0:PRINT "AHORE CARGUE LA AVENTURA TER
MINADA":PRINT "CON:"
10001 PRINT "ENTER";CHR$(34);AVENTURA$
:PRINT :PRINT "QUE SE DIVIERTA!":NEW
```

6. APENDICE

CORRECIONES DE PROGRAMAS

Va a ser imprescindible, que el poseedor del ATARI 600 XL repase los programas publicados en este libro para optimizar la memoria disponible.

Pero también los poseedores de ATARI 800 XL pueden sacar partido de las siguientes propuestas.

Naturalmente, todo lo aquí expuesto puede aplicarse a cualquier programa BASIC.

OPTIMIZACION DE LA MEMORIA

Procure utilizar en lo posible la máxima longitud de cada línea de programa, con lo que se reducirá el número de líneas.

Cada línea de instrucción ocupa varios bytes para su propio número , además de almacenar datos necesarios para encontrar la próxima línea de programa.

¡Si usted utiliza las abreviaciones permitidas, podrá incluso entrar líneas que aparezcan demasiado largas en el listado!

¡Prescinda de todas las referencias REM!

¡Utilice sólo expresiones de una o dos letras en vez de los largos nombres de variables, que nosotros hemos utilizado para facilitarle el manejo del programa!

Las siguientes propuestas ayudan al interpretador Basic a ejecutar sus programas de la forma más rápida posible. Naturalmente, también las propuestas mencionadas anteriormente contribuyen a aumentar la velocidad de ejecución (speed - up).

La única manera de alcanzar este objetivo nos obliga a respetar los intereses del interpretador.

El interpretador confecciona una serie de listas mientras dura su operación. El orden de los elementos de esta lista dependerá del orden en que sean entrados.

De esta forma, y según su colocación en la lista, se asigna a cada variable un espacio en la sección prevista que forma parte de la memoria del programa. Si el ordenador accede ahora a una determinada variable para leer o escribir en ella, deberá repasar todos los elementos de la lista hasta llegar a la variable en cuestión.

Por ello, resulta útil entrar todas las variables por orden de frecuencia, lo que se puede conseguir mediante una línea de programa que le asigne, por ejemplo, el valor cero.

También debería definir como variables las constantes utilizadas con frecuencia, ya que estos valores fijos deben ser transformados en cada operación según el formato de procesamiento del ATARI, hecho que normalmente sólo se da una vez, esto es, al definirlas.

¡Preste especial atención a los bucles!

Nuestro segundo punto de partida hace referencia a los saltos que puede realizar un programa. Pues ¿cómo puede saber el interpretador en que lugar de la memoria se

encuentra el destino del salto?

Muy sencillo, no lo sabe. Se limita a repasar todas las líneas hasta encontrar la línea en cuestión.

Es por ello, que le aconsejamos situar las subrutinas utilizadas con mayor frecuencia al principio del programa. De esta manera se reduce el tiempo de búsqueda, aunque se trate sólo de décimas de segundo.

INDICE ALFABETICO

JUEGOS ESTRATEGICOS
Y COMO PROGRAMARLOS EN EL
ATARI 600XL/800XL/130XE

Una lograda introducción al sugestivo tema de los "juegos estratégicos". Desde juegos sencillos con estrategia fija a juegos complejos con procedimientos de búsqueda hasta programas con capacidad de aprendizaje - muchos ejemplos interesantes, escritos por supuesto de forma facilmente comprensible. Con programas de juegos ampliamente detallados: NIM con un montón, bloqueo, hexapawn, mini-damas y muchos más.

Schneider
Juegos estratégicos - y como programarlos en el ATARI 600XL/800XL/130XE
181 páginas, 1600 .- ISBN 84-86437-14-8

EL LIBRO ESCOLAR PARA
ATARI 600XL/800XL/130XE

Muchos programas interesantes de soluciónes de problemas y de aprendizaje, descritos de forma amplia y comprensible, y adecuados sobre todo para escolares. ¡Aquí el aprendizaje intensivo se convierte en una tarea divertida! Al márgen de temas como los verbos irregulares, o las ecuaciones de segundo grado, un resúmen corto de las bases del tratamiento electrónico de datos, y una introducción a los principios del análisis de problemas, completan este libro que debería obrar en posesión de cualquier escolar.

Voss
El libro escolar para ATARI 600XL/800XL/130XE
389 páginas, 2.800 .- ISBN 84-86437-12-1

Puesta al día de datos

Sello librero

EDITORIAL DATA BECKER S.A. mantiene vivo y amplía el contenido informativo de sus libros y programas, mediante el envío de un servicio de puesta al día, junto con una síntesis noticiosa de la actualidad y perspectivas de la realidad informática en lengua castellana.
Agradecemos cualquier sugerencia o crítica que desee formular y que nos ayude a mejorar las ediciones. Muchas gracias.

¿Qué añadiría? .....

.....

¿Qué suprimiría? .....

.....

Observaciones .....

.....

Título del libro .....

Nombre .....

Dirección ..... Tfno. .....

Código Postal y Población ..... Provincia .....

UN SERVICIO GRATUITO

Sello librero

Información y Pedidos

DATA BECKER S.A. cuenta con un amplio fondo de libros y Software y mantiene un servicio de información por correo sobre las novedades que edita.
Agradecemos nos indique los temas que representan para Vd. mayor interés.

Libros

- ☒ ATARI
- ☐ COMMODORE
- ☐ SINCLAIR
- ☐ MSX
- ☐ LENGUAJES
- ☐ IBM
- ☐ AMSTRAD
- ☐ APPLE
- ☐ SOFTWARE

Si está interesado en recibir alguno de estos servicios, rellene y envíe la tarjeta correspondiente; no necesita franqueo. Muchas gracias.

DESEO RECIBIR EL LIBRO .....

EL PROGRAMA .....

☐ Adjunto cheque ☐ Contra reembolso

Nombre .....

Dirección ..... Tfno. .....

Código Postal y Población ..... Provincia .....

UN SERVICIO GRATUITO

RESPUESTAS POSTALES
PAGADAS

El franqueo será pagado por el Destinatario

DATA BECKER S.A.

APARTADO ESPECIAL N° 4
1448 – SUCURSAL 48 (B)
BUENOS AIRES – ARGENTINA

RESPUESTAS POSTALES
PAGADAS

El franqueo será pagado por el Destinatario

DATA BECKER S.A.

APARTADO ESPECIAL N° 4
1448 – SUCURSAL 48 (B)
BUENOS AIRES – ARGENTINA

Este libro se terminó de imprimir en el mes
de Junio de 1987 en la Planta Impresora
de Edicient S.A.I.C., cita en Mario Bravo 465/71
(1870) Avellaneda - Pcia. Buenos Aires